

DECnet-DOS

digital

Network Management Guide

Network Management Guide

Order Number: AA-PAFHA-TK

DECnet-DOS

Network Management Guide

November 1989

This manual describes the DECnet-DOS Network Control Program (NCP). It explains how you can use NCP commands to set up your personal computer as a node in a DECnet network. It also explains how to use NCP commands to obtain detailed information about your node and the network.

Supersession/Update Information: This is a new manual.

Operating System and Version: MS-DOS V3.1
MS-DOS V3.2
MS-DOS V3.3
MS-DOS V4.0

Software Version: DECnet-DOS V3.0

digital™

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.


The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital or its affiliated companies.

Copyright © 1989 by Digital Equipment Corporation
All Rights Reserved.
Printed in U.S.A.

The postage-prepaid Reader's Comments form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DECnet-DOS	VAX
DECmate	DECUS	VAXcluster
DECnet	ThinWire	VMS
	UNIBUS	VT

IBM is a registered trademark of International Business Machines Corporation.

PC/XT and Personal Computer AT are trademarks of International Business Machines Corporation.

MS, MS-DOS, and MS OS/2 are registered trademarks of Microsoft Corporation.

3Com is a trademark of 3Com Corporation.

MICOM-InterLan is a registered trademark of MICOM Systems, Inc.

Contents

Preface

1 Overview of DECnet-DOS Network Management

1.1	About DECnet-DOS Networks	1-1
1.1.1	DECnet-DOS Terminology	1-2
1.2	Managing a DECnet Node—the Network Control Program	1-4
1.3	Running the NCP Utility	1-5
1.3.1	General Command Format	1-6
1.3.2	Getting HELP	1-8
1.3.3	Exiting from NCP	1-9
1.4	Network Management Servers	1-9
1.4.1	The Loopback Mirror (MIRROR)	1-9
1.4.2	Using the Network Management Listener (NML)	1-9
1.5	Restarting a DECnet-DOS Node	1-10
1.6	DECnet Network Management Databases for DOS	1-10

2 Configuring a DECnet-DOS Node

2.1	Setting Up Your Node	2-1
2.1.1	Naming Remote Nodes	2-3
2.1.2	Specifying Default Access Control Information	2-3
2.1.3	Specifying Default Remote Host Type	2-4

2.1.4	Displaying Default Access Control Information	2-5
2.2	Understanding the DECnet Network Management Databases	2-6

3 Operating a DECnet-DOS Executor Node

3.1	Displaying Network Information for Your Node	3-1
3.1.1	Displaying Default Access Control Information	3-2
3.1.2	Displaying Executor Node Information	3-3
3.2	Changing Network Information for Your Node	3-5
3.2.1	Line State	3-6
3.2.2	Buffer Size	3-6
3.2.3	Segment Buffer Size	3-6
3.2.4	Maximum Links	3-6
3.2.5	Executor State	3-6
3.2.6	Node Name and Node Address	3-7
3.3	Naming Remote Nodes	3-7
3.3.1	Specifying Default Access Control Information for a Remote Node	3-8

4 Monitoring a DECnet Network

4.1	Monitoring Your Node in a DECnet Network	4-1
4.1.1	General Counters	4-2
4.1.2	Line Counters	4-2
4.1.3	Circuit Counters	4-3
4.1.4	Executor Counters	4-3
4.1.5	Displaying Counter Information	4-4
4.1.6	Displaying Default Access Control Information	4-6
4.1.7	Displaying Executor Node Information	4-7
4.1.8	Displaying Remote Node Information	4-8
4.1.9	Displaying Line Information	4-9
4.1.10	Displaying Circuit Information	4-10
4.1.11	Monitoring Events on Your Local Node	4-11
4.2	Monitoring Other Nodes on the DECnet Network	4-11
4.3	NCP Commands Supported by DECnet-DOS NML	4-12

5 Testing the Network

5.1	Overview of the NCP Test Commands	5-1
5.2	Testing the Network with Loopback Messages	5-2
5.2.1	Loopback Counters	5-6

6 NCP Commands for DECnet-DOS

6.1	NCP Commands	6-1
	CLEAR	6-6
	CLEAR EXECUTOR	6-7
	CLEAR LOCAL-ADAPTER-NAME	6-9
	CLEAR REMOTE-ADAPTER-NAME	6-11
	COPY NODE	6-12
	DEFINE	6-14
	DEFINE ACCESS	6-15
	DEFINE CIRCUIT	6-17
	DEFINE EXECUTOR	6-19
	DEFINE LINE	6-23
	DEFINE NODE	6-25
	DEFINE OBJECT	6-27
	DEFINE REMOTE-ADAPTER-NAME	6-30
	EXIT	6-32
	HELP	6-33
	LIST	6-34
	LIST ACCESS	6-35
	LIST CIRCUIT	6-37
	LIST EXECUTOR	6-39
	LIST LINE	6-41
	LIST NODE	6-42
	LIST OBJECT	6-45
	LIST REMOTE-ADAPTER-NAME	6-47
	LOOP	6-49
	LOOP CIRCUIT	6-50
	LOOP EXECUTOR	6-54
	LOOP LINE CONTROLLER	6-56

LOOP NODE	6-57
MIRROR	6-60
MONITOR LOGGING	6-61
PURGE	6-62
PURGE ACCESS	6-63
PURGE EXECUTOR	6-65
PURGE NODE	6-67
PURGE OBJECT	6-69
PURGE REMOTE-ADAPTER-NAME	6-70
READ LOG	6-71
SET	6-73
SET CIRCUIT	6-74
SET ECHO	6-76
SET EXECUTOR	6-78
SET LINE	6-82
SET LINK	6-84
SET LOCAL-ADAPTER-NAME	6-86
SET PAUSE	6-87
SET REMOTE-ADAPTER-NAME	6-89
SHOW ACTIVE	6-91
SHOW ADJACENT	6-93
SHOW CIRCUIT	6-94
SHOW EXECUTOR	6-96
SHOW LAT	6-98
SHOW LINE	6-99
SHOW LOCAL-ADAPTER-NAME	6-101
SHOW NODE	6-102
SHOW REMOTE-ADAPTER-NAME	6-104
TELL	6-105
ZERO	6-108
ZERO CIRCUIT	6-109
ZERO EXECUTOR	6-110
ZERO LAT	6-111
ZERO LINE	6-112

A NCP Error Messages

A.1	Introduction	A-1
A.2	Types of NCP Error Messages	A-1
A.3	Command Line Errors	A-1
A.4	File Input/Output (I/O) Errors	A-4
A.5	Network Error Messages	A-4
A.6	Loop Test Error Messages	A-6

Glossary

Index

Examples

1-1	An NCP Command	1-6
-----	--------------------------	-----

Figures

1-1	A Sample Network	1-3
4-1	Known Remote Node Names	4-6
5-1	Loop Executor	5-4
5-2	Loop Line Controller	5-4
5-3	Loop Node	5-5
5-4	Loop Circuit	5-6

Tables

6-1	NCP Command Summary	6-1
-----	-------------------------------	-----

1
2
3
4
5
6
7
8
9
10

11
12
13
14
15
16
17
18
19
20

21
22
23
24
25
26
27
28
29
30

31
32
33
34
35
36
37
38
39
40

41
42
43
44
45
46
47
48
49
50

Preface

The *DECnet-DOS Network Management Guide* provides the information necessary for you to manage a DECnet-DOS™ system within a DECnet™ network. This guide defines and explains the terms and concepts related to DECnet network management, and describes the network management command set for DECnet-DOS Version 3.0.

The term “DECnet-DOS” refers to DECnet-DOS Version 3.0 running on supported industry-compatible personal computers using the MS-DOS® operating system.

For a list of all supported industry-compatible personal computers and supported operating systems, see the *DECnet PCSA Client for DOS V3.0 Software Product Description*.

Objectives of this Manual

The *DECnet-DOS Network Management Guide* describes the Network Control Program (NCP). An overview section describes DECnet network management concepts and defines network management terms. The guide explains how you can use NCP commands to set up your personal computer as a node in a DECnet network. It also explains how to use NCP to obtain detailed information about your node and the network. In addition, it includes information about loopback testing, event logging, and displaying system counters and database information.

Intended Audience

This guide is intended for users who need to set up, monitor, update, and troubleshoot their DECnet-DOS node in a DECnet network. You should be familiar with the MS-DOS operating system, DECnet network concepts, and DECnet testing procedures (such as loopback).

How to Use This Manual


This manual consists of six chapters, one appendix, and a glossary.

- Chapter 1 introduces DECnet-DOS network management and the Network Control Program (NCP) and explains how to use the NCP commands. This chapter discusses your personal computer's relationship to the DECnet network, components and characteristics of a DECnet network, and functions of the DECnet-DOS network management utility.
- Chapter 2 describes the NCP commands needed to set up the DECnet-DOS network software on your personal computer.
- Chapter 3 describes the NCP commands needed to maintain and update the network software on your personal computer.
- Chapter 4 describes the NCP commands needed to monitor the network software on your personal computer. Monitoring the software on other nodes is discussed in the description of the Network Management Listener (NML).
- Chapter 5 describes the NCP commands that you can use to test your personal computer's network hardware and software. The chapter discusses the use of the loopback mirror in loop testing.
- Chapter 6 describes the DECnet-DOS network management command set and its use. The commands are discussed in alphabetical order.
- Appendix A lists the error messages you can receive while using NCP.
- The Glossary contains a list of DECnet-DOS network management terms.

If you are familiar with networking concepts and DECnet products, you can omit reading the overview in Chapter 1. If you are familiar with network management and loop testing, you can go directly to Chapter 6 for the command descriptions. These command descriptions are presented in alphabetical order with a summary at the beginning of the chapter.

Graphic Conventions Used in This Document

The following table defines the graphic conventions used in this manual:

Convention	Meaning
Special Type	Indicates examples of system output or user input. System output is in black; user input is in red.
COMMAND	Bold characters represent acceptable abbreviations for commands. For example, DELETE indicates that the acceptable abbreviation for the DELETE command is DEL.
UPPERCASE	In commands and examples indicates that you should enter the exact characters shown. However you can enter them in either uppercase or lowercase.
<i>italics</i>	In commands and examples indicates a value that either the system supplies or you should supply.
{ }	Braces indicate that you are required to specify one (and only one) of the enclosed options. Do not type the braces when you enter the command.
[]	Square brackets indicate that the enclosed data is optional. (If a vertical list of options is enclosed, you can specify only one option.) Do not type the brackets when you enter the command.
()	Parentheses enclose a set of options that must be specified together.
vertical list of options	A vertical list of options not enclosed within braces, brackets, or parentheses indicates that you may specify any number of options (or in some cases none if defaults apply).
key	Indicates that you should press the specified key. CTRL/x indicates that you should hold down the CONTROL key while you press the x key, where x is a letter.
Return	Indicates that you should press the key that executes commands. This key may be labeled Enter , Return , or  , depending on your keyboard.

Associated Documents

The following documents are included in the DECnet-DOS documentation set:

- *DECnet-DOS Getting Started*

This manual provides an overview of DECnet-DOS. It also includes an explanation of the product set, a road map through the documents, a brief description of DECnet-DOS utilities and the tasks that you perform with them, and an introduction to frequently used DECnet-DOS commands.

- *Installing DECnet PCSA Client for DOS (with Diskettes)*

This guide describes procedures for installing and verifying the DECnet-DOS software on selected personal computers. This guide addresses a varied audience by providing the following:

- Instructions for using the automated installation procedure for the nontechnical end-user.
- Reference material for the technically oriented user who wants to bypass the automated system prompting.

- *DECnet-DOS User's Guide*

This manual describes the DECnet-DOS utilities and tells how to use them. It also provides an overview of network concepts and defines network terms. The DECnet-DOS utilities discussed in the user's guide include:

Job Spawner (Spawner)

File Access Listener (FAL)

Network File Transfer (NFT)

Network Device Utility (NDU)

Mail System (MAIL)

- *DECnet-DOS SETHOST Terminal Emulation Guide*

This guide describes the SETHOST network virtual terminal utility. It explains how to use SETHOST and the set-up screens to connect to a host node and emulate a terminal connected to that node.

This guide also describes how to use scripts, which are text files containing commands that allow SETHOST to perform many operations automatically.

- *DECnet-DOS Programmer's Reference Manual*

This manual documents the programming interface and language library provided in the DECnet-DOS kit. This manual contains three parts: Part I provides a tutorial on writing network applications; Part II details the programming utilities and network programming calls used in the creation of DECnet-DOS application programs; Part III provides reference information in six appendixes.

You should have available the installation guide and manuals for the MS-DOS operating system and your personal computer.

It is important that you also read the contents of the file README.TXT, which is included on the first floppy diskette of your installation kit.

THE UNIVERSITY OF CHICAGO
DIVISION OF THE PHYSICAL SCIENCES
DEPARTMENT OF CHEMISTRY

REPORT OF THE
COMMISSIONER OF THE
BUREAU OF CHEMISTRY
FOR THE YEAR 1900
CONTAINING
A SUMMARY OF THE
WORK OF THE BUREAU
DURING THE YEAR
AND A LIST OF THE
PUBLICATIONS OF THE
BUREAU FOR THE YEAR

CHICAGO
PUBLISHED BY THE
UNIVERSITY OF CHICAGO PRESS
1901

Overview of DECnet-DOS Network Management

This book describes the DECnet-DOS™ Network Management utility. To use the network management tools effectively, you should be familiar with the concepts and terminology of networking and network management. This chapter will familiarize you with network management by discussing the following topics:

- An overview of network management concepts
- Running NCP
- General NCP command formats
- Getting help
- Using the Network Management Listener
- An overview of data storage

You should know the person responsible for overall management of your network. If you have questions regarding your node or the network, this person should be able to provide you with extra information or assistance.

1.1 About DECnet-DOS Networks

A personal computer, has a limited amount of space to store information. This capacity is increased by the use of removable diskettes and backup devices. However, the information stored on these media is not quickly accessible, and you are still limited to information contained on your system.

A network is two or more computers connected together via communications lines (see Figure 1-1). When it is part of a network, your personal computer can access additional information including information from remote locations

DECnet-DOS software creates an extension to your MS-DOS® operating system enabling it to interact in a network environment. DECnet-DOS commands allow you to exchange information over the network communication lines. With DECnet-DOS you can:

- Send mail.
- Transfer files between your system and another on the network.
- Print a file on a printer connected to another system on the network.
- Use a disk attached to another computer on the network as if it were a disk on your own personal computer.
- Connect to a remote system using your node as if it were a terminal.

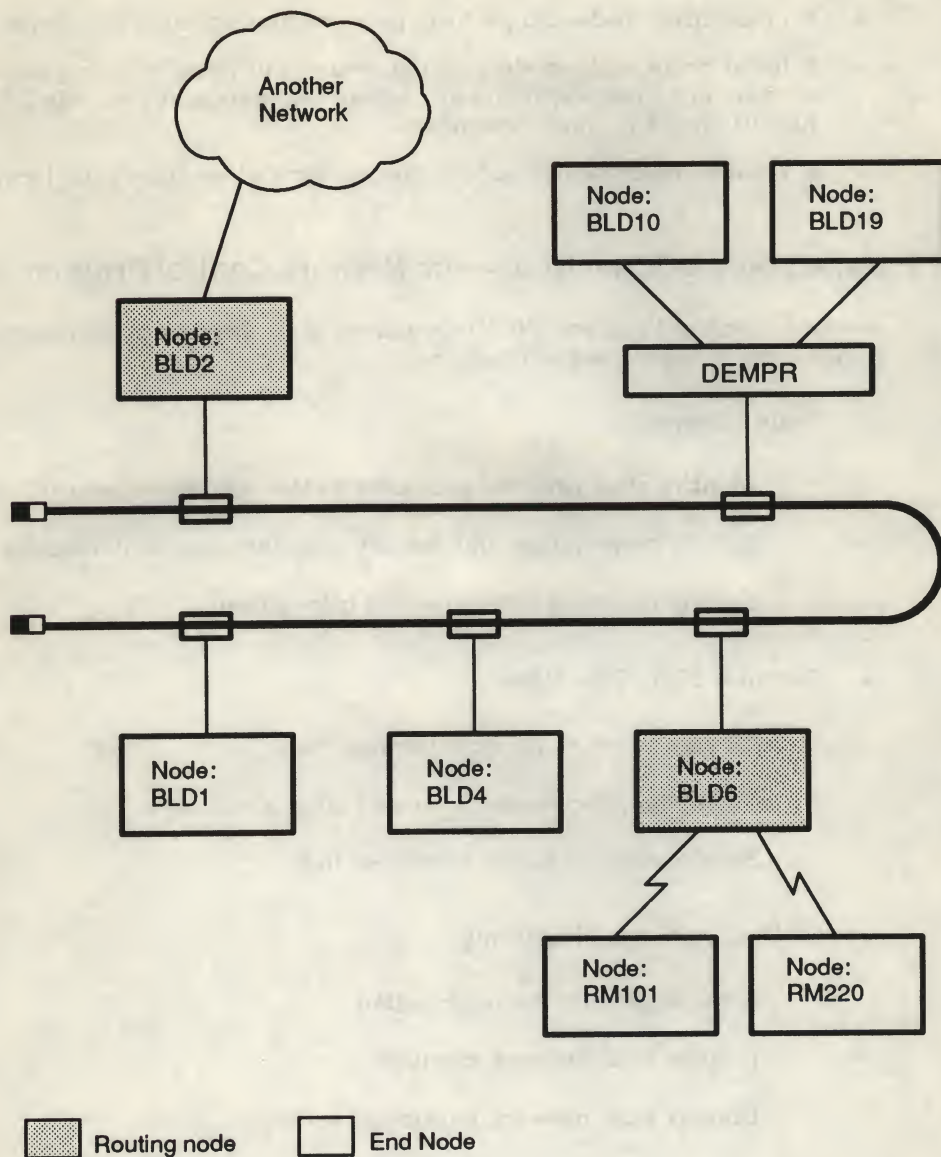
1.1.1 DECnet-DOS Terminology

As shown in Figure 1-1, personal computers are not the only components connected to the network. The DECnet™ software allows connections to any Digital Equipment Corporation system, many personal computers, and with the appropriate hardware and software configurations, connections to SNA networks and Packet Switched Data Networks (PSDNs). Each of these component systems on a DECnet network is called a **node**.

Throughout this guide the term “node” refers to any processor, intelligent terminal or other computer system capable of functioning independently within a DECnet network environment. Figure 1-1 depicts a typical network to illustrate terms and further define how nodes are used in the context of a DECnet network:

- An **adjacent node** is physically connected to another node by a single communications line, as BLD1 and BLD4.
- An **end node** such as BLD10, BLD19, BLD1, BLD4, RM101, or RM220 can receive information for its own use; it cannot receive messages and subsequently route them to another node.

Figure 1-1: A Sample Network



LKG-3162-891

- A **routing node**, such as the shaded nodes BLD2 and BLD6 can receive and forward information from one node in the DECnet network to another node or network.
- An **executor node** can perform network management functions.
- A **local node** is the node at which you are physically typing commands. For instance, if you are typing commands at the node RM101, RM101 would be your local node.
- A **remote node** is any node in the network other than your local node.

1.2 Managing a DECnet Node—the Network Control Program

Network Control Program (NCP) commands from your personal computer perform the following set of functions:

- **Node Generation**

Identify your personal computer to the rest of the network

Specify node names and default outgoing access information

Specify incoming access control information

- **Executor Node Operation**

Change network services parameters

Turn your executor node on and off in a network

Set the state of a line, circuit, or link

- **DECnet Network Monitoring**

Read or monitor log information

Display local network counters

Display local network parameter settings

- **Node Diagnostics and Testing**

Display information about your node and how it is interacting with the network

Send test messages over each element of a connection

1.3 Running the NCP Utility

You can run the NCP utility using one of the following three methods:

- **Single command method:**

Enter NCP followed by the specific NCP command. Press **Return**. After you enter each NCP command, the system returns the DOS prompt. For example:

```
C:\>NCP command Return  
C:\>
```

- **Single command method using a command or redirected input file:**

Use this method for entering commands from a command file. A command file (also called a redirected input file) contains all the NCP commands to complete a series of NCP operations. By using a command file, you need issue only one command to process all other commands in the file.

To enter commands using a command file, type NCP followed by a left angle bracket (<) and the name of the file. Then press **Return**. For example:

```
C:\>NCP <file-name Return
```

NCP returns an error level value on completion. This error level can be used by the IF ERRORLEVEL statement in command files. If NCP has run successfully, it returns a value of 0. If the execution was not successful, it returns a non-zero value.

By using the SET ECHO command, you can specify whether a redirected input file is displayed on your screen as the file is executed. See Chapter 6 for specific information on how to turn echo on and off.

- **Multiple command method:**

Supply the name of the utility (NCP) and press **Return**. NCP responds with its own prompt. (You cannot use this method for submitting NCP commands in redirected mode.)

C:\>NCP Return
NCP>

Enter the NCP command after the NCP prompt and press Return.
Continue this procedure until you have entered all your commands.

NCP>command Return
NCP>command Return
NCP>command Return
NCP>

1.3.1 General Command Format

Most NCP commands consist of three parts: the command verb, an entity on which the command operates, and one or more parameters that further qualify the action to be taken. For each command, you must supply the verb, one entity and (usually) one or more parameters or qualifiers.

The following example shows parts of an NCP command:

Example 1-1: An NCP Command

NCP>SHOW EXECUTOR CHARACTERISTICS TO EXECUTOR.LIS Return

VERB	ENTITY	PARAMETER	QUALIFIER
------	--------	-----------	-----------

You can abbreviate commands and keywords to three or more unique characters. (An exception to this is the use of the keywords **LINK** and **LINE**. The first three characters for each keyword are the same; so you must enter all four characters to properly identify the keyword.) For example, you can type **SHO** for **SHOW**. However, for clarity and consistency, all examples in this manual show the full command format.

The following are NCP command verbs:

CLEAR	removes or clears temporary parameters.
COPY	COPY TO copies the node database to another file (without the access information). COPY FROM copies node information from another node into your database.

DEFINE	establishes specific parameters that take effect when you perform a network restart.
EXIT	simply exits from NCP.
HELP	provides on-line information about the NCP utility and all of its commands.
LIST	displays information set up using the DEFINE command. You can also redirect this information to an output file.
LOOP	runs loopback tests that let you check the following: <ul style="list-style-type: none"> ■ the operation of your local node ■ the connection to the network ■ the connection to a remote node on the network ■ the communication hardware (including modems) that connects your system to the network
MIRROR	sets up a software loopback mirror to echo messages from a remote node back to that remote node. Before a remote node can run a LOOP test to your node, you must first start the mirror on your local node.
MONITOR	continuously displays event logging information as it occurs.
PURGE	deletes information from the DECnet databases.
READ	displays event logging information. You can also redirect log information to a specified output file.
SET	establishes specific parameters for the network. These parameters (and their values) are not saved across network restarts.
SHOW	displays information about the currently running system. This information includes counter displays, parameters established with the SET command, and parameters that cannot be set. You can redirect information displayed by the SHOW command to a specific output file.
TELL	instructs a remote node to display information about its executor, lines, or circuits.
ZERO	resets the counters associated with a specific network entity.

1.3.2 Getting HELP

If you need assistance in selecting NCP commands, type HELP at the NCP> prompt; then press **Return**. For example:

```
NCP>HELP Return
```

The system responds with the following display:

The HELP command displays information about NCP commands on your screen. HELP is available for the following commands:

;	CLEAR	COPY	DEFINE	EXIT	LIST	LOOP	MIRROR
MONITOR	PURGE	READ	SET	SHOW	TELL	ZERO	

You can type the name of a command for which you want information. For example:

```
NCP>HELP CLEAR Return
```

The system responds with the following display:

HELP is available for the following commands:

CLEAR EXECUTOR	CLEAR KNOWN
CLEAR LOCAL-ADAPTER-NAME	CLEAR REMOTE-ADAPTER-NAME

You can get additional help on a particular NCP command. For example, if you want more information on the CLEAR EXECUTOR command, type the following command:

```
NCP>HELP CLEAR EXECUTOR Return
```

The system responds with the following display:

Removes specified executor node parameters from running system (DDCMP only).

RECEIVE PASSWORD Password the executor node must receive from the adjacent node in order to exchange messages.

TRANSMIT PASSWORD Password the executor node must transmit to the adjacent node in order to exchange messages.

1.3.3 Exiting from NCP

To exit from NCP, you can use one of two methods. You can use the NCP EXIT command, or you can enter **Ctrl/Z** and then press **Return**. The currently running command is completed, then NCP exits.

For example:

```
NCP>EXIT Return  
OR
```

```
NCP>Ctrl/Z Return
```

You can also use **Ctrl/C** to leave NCP. If you use **Ctrl/C**, however, the currently running command is interrupted and might not complete.

1.4 Network Management Servers

NCP uses various servers to handle requests from remote systems. These servers enable other nodes to request information from your local node. Likewise, the same servers, running on remote nodes, allow you to gather status from these remote nodes using NCP.

1.4.1 The Loopback Mirror (MIRROR)

You can use the MIRROR command, detailed in Chapter 6, to echo loopback messages to the remote nodes that issue them. The loopback mirror runs in the foreground of DECnet-DOS nodes, so no other activity can take place while this server is running.

When you issue loopback messages to a remote node, the MIRROR returns them. If no message is returned from the remote node MIRROR, a problem must exist between your local node and the remote node to which you sent the loopback message.

1.4.2 Using the Network Management Listener (NML)

The Network Management Listener (NML) is a background process that responds to network management queries from remote DECnet systems. These systems must be running NCP or other network management tools that use the NICE V4.0 protocol.

If you want to zero the circuit counters on another personal computer on the DECnet network, you can issue the following command:

```
NCP> TELL BLD10 ZERO CIRCUIT COUNTERS Return
```


This NCP command would tell the node BLD10 to zero all of its circuit counters. If NML is running on BLD10, it will zero all of its circuit counters. If NML is not running on BLD10, an error message will be displayed on the requesting node.

NML runs in the background so that it does not interfere with normal use of your personal computer. As a background task, it is limited to examining the currently running system (the volatile databases, whose information is displayed by using the SHOW commands). NML cannot access any information residing in the file system (the permanent database whose information is displayed by using the LIST commands).

1.5 Restarting a DECnet-DOS Node

DECnet-DOS V3.0 introduces the new capability of stopping and restarting network software. The effect of stopping and restarting network software is much the same as the effect of rebooting your DECnet-DOS node. Your node will refer to permanent databases for all default networking information, and any network states will be reset to the values in the permanent databases. To stop the network process, simply unload the network software (see *Memory Solutions*). To restart the network, simply reload the network software.

The term **network restart** encompasses both rebooting a running DECnet node and restarting stopped network software.

1.6 DECnet Network Management Databases for DOS

All DECnet nodes use a similar system of data organization. That is, all DECnet nodes use a system of **permanent** and **volatile** databases to configure component characteristics.

Permanent databases store on disk the characteristics you want your system to use when DECnet-DOS is started. Permanent databases are retained across network restarts.

Volatile databases are stored only in the running DECnet process in your computer's memory. They contain the values that you have used since the time DECnet-DOS was started. They are not stored in the permanent databases because you might not want to use them again. You can choose whether to store characteristics permanently. Volatile databases serve as a temporary memory for your computer.

Configuring a DECnet-DOS Node

This chapter provides an overview of the NCP commands and parameters that you can use to set up your node as a DECnet-DOS™ executor node. On an executor node you can run NCP commands to perform network management functions in a DECnet™ network.

2.1 Setting Up Your Node

You must choose a name and address to identify your node to other nodes in the network. Each node must have a unique name and address. In most cases, you obtain your node name and address from a system or network manager in your facility. The person coordinating network activities must keep a list of all node names and addresses to ensure there is no duplication.

Your local node address and name are established when you use the DECnet-DOS installation procedure. You need not repeat this step after installation is completed. *Installing DECnet PCSA Client for DOS (with Diskettes)* provides a step-by-step explanation of the installation procedure.

However, if you need to change your node name and node address, you can change them with the DEFINE EXECUTOR command. The DEFINE EXECUTOR command establishes executor characteristics that include node name and address. The executor's node name and address are stored in the permanent database, DECPARM.DAT, and therefore, become effective when you perform a network restart.

The following command enters the executor name as ROCKY in the permanent database, DECPARM.DAT:

```
NCP>DEFINE EXECUTOR NAME ROCKY ADDRESS 2.975 Return
```

This command will not take effect until the next network restart because it changes an entry in the permanent database. For the change to take effect immediately, type the following command:

```
NCP>SET EXECUTOR NAME ROCKY ADDRESS 2.975 Return
```

NOTE

You must set the executor state to off before changing the executor address.

You should change database information in both the permanent and the volatile databases, using both **DEFINE** and **SET**.

A **node name** is an identification string consisting of one to six alphanumeric characters and including at least one alphabetic character. For example:

RM101

A **node address** is a unique numeric identification of a specific node. The node's address includes an area number and a node number:

area.node

where:

area is a number in the range of 1 to 63.

node can be in the range of 1 to 1023. It is separated from the area number by a period. For example:

2.975

In this example, the area number is 2 and the node number is 975.

No two nodes can have duplicate node addresses in a single DECnet network.

2.1.1 Naming Remote Nodes

At your local node, you can create a list of the remote nodes that you want to access by name. This list can include access control information that allows default access to a specific account on a remote node. (This is the outgoing access control information that is stored in the DECALIAS.DAT database). To specify remote node information, use the DEFINE NODE command.

When you create a list of remote node names, the following information is required:

- **Node address**

A numeric string including the area number in the range of 1 to 63 and the node number in the range of 1 to 1023. If you enter a node address without an area number, the default area number is that of your executor node.

- **Node name**

A character string consisting of 1 to 6 alphanumeric characters with at least one alphabetic character.

For instance, to access node RM2, you should first type the following command:

```
NCP>DEFINE NODE 2.65 NAME RM2 Return
```

Once RM2 is defined, you can access it without having to specify the node number. The number 2.65 is the node address for the node RM2.

2.1.2 Specifying Default Access Control Information

Access control information allows you to access a specified remote node with the privileges of a specified user. This information includes:

- **User name**

A character string consisting of 1 to 39 alphanumeric characters that identifies the user at the remote node.

- **Password**

A character string consisting of 1 to 39 alphanumeric characters that might be required to access files or programs on the remote node.

■ Account

A character string consisting of 1 to 39 alphanumeric characters. (This field is provided for systems that require a field containing information for system accounting. It is not required on all systems.)

To store the characters representing the access information in uppercase letters, type the characters. To store the characters exactly as typed, place the string in double quotation marks. Character strings that are not in quotation marks are forced to uppercase.

For example:

abc is stored in the database as **ABC**

"Aab" is stored in the database as **Aab**

When accessing a remote node by node name only, a DECnet-DOS utility follows this procedure to determine access control information:

1. The utility first checks to see if access information has been provided directly to the utility. If this is true, then the utility uses that information to access the remote node.
2. The utility determines if default access information was specified with an NCP DEFINE NODE command by checking the permanent database file, DECALIAS.DAT.
3. If no default access control information was specified using NCP, the utility assumes that the remote node does not require access control information, and the requested operation is attempted using no access information.

2.1.3 Specifying Default Remote Host Type

You can also specify that the node you want to access is a LAT host node or an SMB server for MS-NET. (The LAT information is used by the SETHOST utility when you connect to a remote node supporting a LAT server.) Use the following designations to specify the appropriate node:

LAT-HOST	Designates this node as a LAT host.
MS-NET	Designates this node as an MS-NET file server.

For example:

```
NCP>DEFINE NODE 2.375 NAME BLD3 USER SMITH PASSWORD JANE LAT-HOST Return
```

This command assigns node name BLD3 to node address 2.375. It specifies the user's name as SMITH, the password as JANE, and the node as a LAT host node.

2.1.4 Displaying Default Access Control Information

The LIST KNOWN NODES command displays a list of remote nodes and any default access control information that has been set for those nodes. If the access control information contains a password, that password is not displayed on the screen. Instead, it is represented by three dots (...).

To display information from the permanent database for the remote nodes known to your local node, type the NCP command, LIST KNOWN NODES. This command displays the following information for nodes that are known to the local node ROCKY:

- Node address
- Node name
- Number of current active logical links to this node
- Access control information

NCP>LIST KNOWN NODES Return

The system responds with a display similar to the following:

```
Node Summary as of 4-MAR-1989 16:36:30
Executor node      = 2.975 (ROCKY)
Executor state     On
Executor Identification DECnet-DOS V3.0

Node   Node   Active  LAT/   Account
Address Name   Links  MS-NET Information
2.375  BLD3    0      M L   /SMITH/...
4.30   BLD10   1
4.60   BLD6    0
4.62   BLD1    0
4.64   RM101    0
55.86  FLR3     0      /DJONES
55.170 FLR5     0
NCP>
```


2.2 Understanding the DECnet Network Management Databases

NCP uses information from six permanent databases. These databases act as storage areas and provide all of the information necessary for NCP to control your DECnet-DOS node. The DECnet-DOS installation procedure creates these databases for you during the installation process. You can then use NCP commands to change or delete specific information in each database file, or you can delete all of the information in a specific database file.

These permanent databases are retained across network restarts and they store the following information:

- Node name
- Node address
- Line characteristics
- Circuit characteristics
- Access information for connections to selected remote nodes
- Access information for connections from remote nodes
- Objects to be run by the Job Spawner
- NETBIOS remote adapter information

To enter, remove, or display information in the permanent database (which will cause values to take effect at the next network restart), use these commands:

DEFINE
PURGE
LIST

NCP also uses volatile databases which are stored in the running DECnet process in your computer's memory. They contain the values that you used since the time DECnet-DOS was started. These values will take effect immediately but they will not be retained across network restarts.

To enter, remove, or display information in the volatile database use these commands:

SET
CLEAR
SHOW

The following are the six permanent databases NCP uses:

- **DECACC.DAT**—This is the permanent incoming access database. You can set up information that allows other systems to access files or programs on your local node. The information is stored on a user-name basis. Applications use the information in this database to check the access privileges of users who are trying to get information from your node. This allows you to control who can or cannot gain access to your node.
- **DECALIAS.DAT**—This is the permanent outgoing access database. You can store information here to be used when you want to access other nodes in the network. Outgoing access information includes a user name, account information, and a password. This information is assigned for each node.
- **DECNODE.DAT**—This is the permanent DECnet node database. It contains a list of node names and corresponding node addresses, a pointer to the outgoing access information, and a LAT or MS-NET indicator.
- **DECOBJ.DAT**—This is the permanent object database that is used by the Job Spawner. The database contains the definitions of the objects that the Job Spawner can access.
- **DECPARM.DAT**—This is the permanent DECnet database. It contains all of the configuration information about your local node, including executor, line, and circuit information.
- **DECREM.DAT**—This is the permanent remote adapter name database. This database is used by MS-NET. It contains information that includes both a remote node name and an object associated with the adapter name.

NOTE

When you start DECnet-DOS, the network process checks for a DECPARM.DAT file. If the file does not exist, the network process initializes using its own set of default parameters.

When you start NCP, the utility also checks for a DECPARM.DAT file. If the file does not exist, NCP uses the default parameters set up by the running network process to create the file. You can then change or modify these parameters by using NCP commands.

If you choose not to set parameters by using NCP, DECnet-DOS will use default values where appropriate. When you boot your system, these values are loaded into areas of your computer's memory where they are readily accessible.

Although you can use both DEFINE and SET to modify network parameters, each command has a different result when you issue it:

- The DEFINE and PURGE commands establish and delete specific parameters, but the changes do not take effect until the next time you perform a network restart. The parameters are stored in the permanent databases: DECPARM.DAT, DECNODE.DAT, DECALIAS.DAT, DECACC.DAT, DECREM.DAT, and DECOBJ.DAT. Values that you have changed with DEFINE and PURGE will be carried across network restarts.
- The SET and CLEAR commands also establish and remove specific parameters, but the changes take effect as soon as the command is issued. These temporary parameters are stored in the volatile database. The parameters established by the SET commands do not retain their values across a network restart unless you also modify them in the permanent database using DEFINE or PURGE.

For detailed information about the DECnet-DOS database manipulation commands, refer to Chapter 6.

Operating a DECnet-DOS Executor Node

This chapter provides an overview of the NCP commands and parameters you can use to maintain and operate your node in a DECnet-DOS™ network. Node maintenance includes the following functions:

- Displaying network information
- Displaying executor node information
- Changing network information for your node
- Naming remote nodes
- Specifying and changing default access control information

3.1 Displaying Network Information for Your Node

Using NCP commands, you can display the following local node and network information on your screen:

- Node, line, and circuit parameters.
- Node names, node addresses, and access control information for remote nodes (if they have been defined locally).
- Counters on the flow of network messages to and from your node. The counters also record error conditions.
- Network status (state) and other related information for a specific node, line, link, or circuit.

In DECnet-DOS™, a circuit is the communications path established between the local node and its adjacent node. A line is the physical line connecting the local node to the adjacent node.

You can use NCP commands to change the executor node, line, and circuit parameters according to the needs of your network. Do not change the settings for these parameters unless you are familiar with DECnet network concepts and are aware of how all nodes (and their parameters and characteristics) interact in the network.

3.1.1 Displaying Default Access Control Information

The NCP SHOW KNOWN NODES command displays a list of remote nodes in the volatile database. Along with the node names, SHOW KNOWN NODES displays the default access information for nodes that have had connections to or from the local node since the last time the network was restarted. Enter the following command:

```
NCP>SHOW KNOWN NODES Return
```

The system responds with a display similar to the following:

Known Volatile Nodes as of 4-Aug-1989 16:36:30

Node Address	Node Name	Active Links	LAT/ MS-NET	Account Information
2.375	BLD10	0		/SMITH/...
4.64	RM5J	0		
55.86	RM101	0		/DJONES
55.335	BLD2	0		

NCP>

To display permanent database information for the remote nodes known to your node, type the NCP command LIST KNOWN NODES.

```
NCP>LIST KNOWN NODES Return
```

This command displays the following information for nodes that are known to the local node ROCKY:

- Node address
- Node name

- Number of current active logical links to this node
- Whether the node is a LAT host or an MS-NET server
- Access control information

The system responds with a display similar to the following:

Known Permanent Node Summary as of 17-FEB-1989 16:36:30

Executor node = 2.975 (ROCKY)

State On
 Executor Identification DECnet-DOS V3.0

Node Address	Node Name	Active Links	LAT/ MS-NET	Account Information
2.375	BLD3	0		/SMITH/...
4.30	RM2	1	M L	
4.60	FLR19	0		
4.62	RM1	0		
4.64	BLD1	0		
4.215	RM10	0		
4.216	RM1A	0		
4.298	FLR1	0		
55.86	FLR3	0		/DJONES
55.124	BLDJ	1	M L	
55.170	FLR5	0		
55.242	BLDK	0	L	
55.261	RM110	0		
55.335	RM5J	0		

NCP>

3.1.2 Displaying Executor Node Information

To display information about the currently running executor node (your local node), enter the following command:

NCP>SHOW EXECUTOR CHARACTERISTICS Return

This command summarizes the network parameters for the currently running executor node as follows:

Executor Characteristics as of 24-FEB-1989 15:21:13

Executor Node	= 2.375
Driver Version Number	= 3.0.0
State	= ON
Executor Identification	= DECNET-DOS V3.0
Executor Version	= 4.2.0
Node Name	= ROCKY
Node Address	= 2.975
Incoming Timer	= 45
Outgoing Timer	= 60
Confidence Timer	= 15
Incoming Proxy	= DISABLED
Outgoing Proxy	= ENABLED
Active Links	= 1
NSP Version	= 4.0.0
Maximum links	= 16
Delay Factor	= 32
Delay Weight	= 2
Inactivity Timer	= 30
Retransmit Factor	= 12
Routing Version	= 2.0.0
Routing Type	= Endnode IV
Maximum Buffers	= 16
Buffer Size	= 1498
Segment Buffer Size	= 557
Receive Password	= (password not set)
Transmit Password	= (password not set)
Database Path	= C:\DECNET\
Transmit pipe Quota	= 6
Receive pipe Quota	= 6
PC Type	= IBM PCAT
Nak Quota	= 0
Autoboot	= ENABLED
NCP>	

To display the executor node values that take effect when you perform a network restart, use the **LIST EXECUTOR CHARACTERISTICS** command, which would produce a display similar to the following:

Executor Characteristics as of 24-FEB-1989 15:21:13

State	= ON
Executor Identification	= DECNET-DOS V3.0
Node Name	= ROCKY
Node Address	= 2.975
Incoming Timer	= 45
Outgoing Timer	= 60
Confidence Timer	= 15
Incoming Proxy	= DISABLED
Outgoing Proxy	= ENABLED
Maximum Links	= 4
Delay Factor	= 32
Delay Weight	= 2
Inactivity Timer	= 30
Retransmit Buffer	= 12
Maximum Buffers	= 16
Buffer Size	= 1498
Segment Buffer Size	= 557
Receive Password	= (password not set)
Transmit Password	= (password not set)
Transmit pipe Quota	= 6
Receive pipe Quota	= 6
Nak Quota	= 0
Autoboot	= ENABLED

NCP>

3.2 Changing Network Information for Your Node

The following sections describe the network parameters that you can change in DECnet-DOS.

NOTE

Some combinations of settings for buffer sizes, buffer counts, and timers can cause performance problems or network failures. Change network parameters only to solve specific problems. If new problems occur as a result, change back to the default settings.

3.2.1 Line State

The state of the communications line at network restart can be either ON or OFF. If a line's state is OFF, it is not available for DECnet activity. Chapter 4 describes network line states and substates in detail.

3.2.2 Buffer Size

Every node in the network uses areas in memory, called buffers, for temporary storage of data. The buffers hold information being transferred to and from each node.

The buffer size parameter defines the maximum number of buffers for communications over LAN links.

This parameter can only be changed using the DEFINE command.

3.2.3 Segment Buffer Size

The segment buffers are used for transmitting DECnet network messages. The segment buffer size defines the maximum packet size for communications over WAN links. Do not change the segment buffer size unless you are very familiar with DECnet network concepts and know how the local network is set up.

You can only change this parameter by using the DEFINE command. Therefore, you must perform a network restart for the change to take effect.

3.2.4 Maximum Links

This parameter defines the maximum number of links available for the node. When you adjust this parameter, the number of buffers is adjusted accordingly. So, the higher the number for maximum links, the more buffers DECnet-DOS reserves.

You must change this parameter using the DEFINE command. Therefore, you must perform a network restart for the change to take effect.

3.2.5 Executor State

When the executor state is ON, remote nodes can establish a logical link to the executor node. If you set the executor state OFF, the DECnet Network Process (DNP.EXE) is deinstalled and the DECnet-DOS software is stopped.

3.2.6 Node Name and Node Address

To change your node name and node address, use the **DEFINE EXECUTOR** command. This command establishes executor characteristics that include node name and address. The executor node name and address become effective when you perform a network restart.

```
NCP> DEFINE EXECUTOR ADDRESS 2.975 NAME ROCKY Return
```

For this value to take effect immediately, use the **SET** command as follows:

```
NCP> SET EXECUTOR ADDRESS 2.975 NAME ROCKY Return
```

NOTE

The executor state must be **OFF** before you set the executor address.

3.3 Naming Remote Nodes

At your local node, you can create a list of the remote nodes to access by name. This list can include access control information that allows default access to a specific account on a remote node. (These node names are stored in the node database, **DECNODE.DAT**.) To specify remote node information, use the **DEFINE NODE** command.

When you create a list of remote node names, the following information is required:

- Node address
- Node name

To define the remote node **BLD3**, type the following command:

```
DEFINE NODE 2.375 NAME BLD3 Return
```

At this point, you might want to know what nodes are already defined. You can do this using the **LIST KNOWN NODES** command.

3.3.1 Specifying Default Access Control Information for a Remote Node

Access control information allows you to access a specified remote node with the privileges of a specified user. (This is the outgoing access control information that is stored in the DECALIAS.DAT database.) This information includes:

- User name
- Password
- Account

To store uppercase letters representing the access information, type the characters. To store the characters as typed, type the string in double quotation marks. Character strings that are not in quotation marks are forced to uppercase.

For example:

abc is stored in the database as **ABC**

"Aab" is stored in the database as **Aab**

Monitoring a DECnet Network

This chapter provides an overview of Network Control Program (NCP) commands and parameters used to monitor your local node or any node in a DECnet™ network. The chapter is divided into two sections: one on monitoring your own node, and the other on monitoring a remote node.

Monitoring consists of the following tasks:

- Observing the current state of local and remote nodes and physical lines.
- Displaying the contents of the permanent and volatile databases for the local and remote systems.
- Inspecting and interpreting the contents of various counters on local and remote nodes.

4.1 Monitoring Your Node in a DECnet Network

DECnet-DOS™ maintains counters of network status at all times. To look into the performance of your node on the DECnet network, refer to these counters for the appropriate information. In addition, you can issue a monitor command that enables you to keep track of network events that occur on your node. This section explains all the information available to you and how to access it.

Observing the following counters can help diagnose certain problems and improve network performance.

4.1.1 General Counters

Seconds since last zeroed

The length of time the counters have been accumulating.

4.1.2 Line Counters

Receive failures

Indicate reception problems. It is typical to have some errors over time. However, many errors over a short period can indicate network hardware configuration problems. (This counter is not accurate for the 3Com™ EtherLink 3C501 driver.)

Receive overruns

Indicate packets lost because the local adapter could not keep up with the network traffic. This is typical for slow systems or single-buffered Ethernet adapters. One way to improve the situation is to disable multicast reception by using the SET CIRCUIT MULTICAST LISTENER DISABLED command. (This counter is not accurate for the 3Com EtherLink 3C501 driver.)

Collision detect check failures

On most systems, this counter indicates that the heartbeat signal is not being properly supplied to the Ethernet device. This counter is used in the 3Com EtherLink 3C501 driver to indicate the "zero length" packet problem.

System buffer unavailable

A packet arrived and no more buffers were available in the data link pool to receive it.

User buffer unavailable

An attempt was made to allocate a buffer for a data link process and none were available in the data link pool.

4.1.3 Circuit Counters

User buffer unavailable

A control message could not be sent on a DEPCA adapter because no more buffers were available in the data link pool. For all other Ethernet adapters, this counter should be zero.

4.1.4 Executor Counters

Response timeouts

Indicates the expiration of a message timeout.

CCB allocation failures

No more internal control blocks were available when requested. The number of CCBs is controlled by the number of links and buffers.

SDB allocation failures

No more Small Data Blocks (SDBs) were available when requested. This occurs if many asynchronous operations are outstanding or if the network is having problems transmitting control messages. The number of SDBs is controlled by the maximum number of links.

LDB allocation failures

No more Large Data Blocks (LDBs) were available when requested. This might occur for Ethernet configurations if the data link buffer pool is at its limit. The number of LDBs is controlled by the EXECUTOR BUFFERS parameter.

This might be because of a delay in acknowledging messages across the network. However, if it appears to affect performance, increase the number of buffers.

Link allocation failures

No more links or sockets were available when requested. The number of links is controlled by the EXECUTOR MAXIMUM LINKS parameter.

Because DECnet-DOS links are not closed when a program is aborted (unless the program does so itself), abort any open network links with the NCP SET KNOWN LINKS STATE OFF command.

Receive connect resource errors

An incoming connect failed because no more links were available. The number of links is controlled by the EXECUTOR MAXIMUM LINKS parameter.

4.1.5 Displaying Counter Information

To view the contents of these counters, use the SHOW command. For example, to see the counters associated with your executor node, type the following command:

```
NCP> SHOW EXECUTOR COUNTERS Return
```

The system responds with a display similar to the following:

Executor Counters as of 1-Aug-1989 15:57:30

Executor node = 2.975 (ROCKY)

Seconds since last zeroed	= 39
User bytes received	= 0
User bytes sent	= 0
User messages received	= 0
User messages sent	= 0
Total bytes received	= 0
Total bytes sent	= 0
Total messages received	= 0
Total messages sent	= 0
Connects received	= 0
Connects sent	= 0
Response timeouts	= 0
Receive connect resource error	= 0
Maximum logical links active	= 1
Packet format error	= 0
Verification error	= 0
CCB allocation failure	= 0
SDB allocation failure	= 0
LDB allocation failure	= 0
Link allocation failure	= 0

NCP>

Type the SHOW LINE COUNTERS command, to see a display like the following:

Line Counters as of 1-Aug-1989 16:03:48

Line = ETHER-1

Seconds since last zeroed	= 763
Bytes received	= 1346481
Bytes sent	= 1980
Data blocks received	= 12477
Data blocks sent	= 33
Multicast bytes received	= 899219
Multicast blocks received	= 9135
Blocks sent, initially deferred	= 0
Blocks sent, single collision	= 0
Blocks sent, multiple collisions	= 0
Send failure	= 0
Receive failure	= 0
Unrecognized frame destination	= 0
Data overrun	= 0
System buffer unavailable	= 0
User buffer unavailable	= 0
Collision detect check failure	= 0

NCP>

Type the SHOW CIRCUIT COUNTERS command, to see a display like the following:

Circuit Counters as of 1-Aug-1989 16:26:44

Circuit = ETHER-1

Seconds since last zeroed	= 2097
Terminating packets received	= 0
Originating packets sent	= 0
Circuit down	= 0
Initialization failure	= 0
Bytes received	= 35053
Bytes sent	= 2310
Data blocks received	= 252
Data blocks sent	= 70
User buffers unavailable	= 0

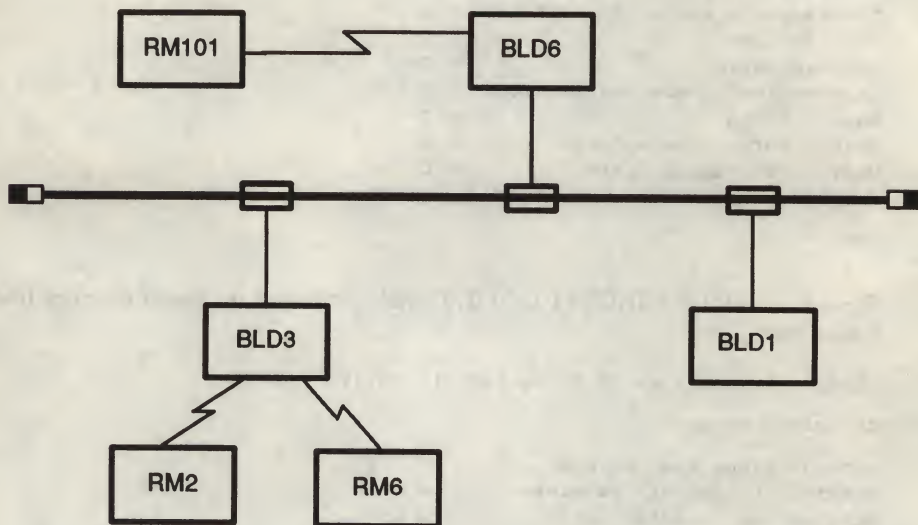
NCP>

4.1.6 Displaying Default Access Control Information

The NCP SHOW KNOWN NODES command displays a list of recently accessed remote nodes and any associated default access control information. If the access control information contains a password, that password is not displayed on the screen. The password is represented by three dots (...).

Figure 4-1 shows remote nodes in a sample DECnet-DOS network. Each node has a specific node name.

Figure 4-1: Known Remote Node Names



LKG-3210-891

To display information for the remote nodes that appear in Figure 4-1, type the NCP command SHOW KNOWN NODES. This command displays the following information for nodes that are known to the local node ROCKY:

- Node address
- Node name
- Number of current active logical links to this node

- Delay
- Next node

NCP>SHOW KNOWN NODES Return

The system responds with a display similar to the following:

Known Volatile Nodes as of 4-Feb-1989 16:36:30

Node Address	Node Name	Active Links	Delay	Next Node Address	Name
2.975	ROCKY	0	5		
4.64	BLD1	0			
55.86	FLR3	0			
55.335	RM5J	0	1		

NCP>

4.1.7 Displaying Executor Node Information

To display information about the currently running executor node, enter the following command:

NCP>SHOW EXECUTOR CHARACTERISTICS Return

The resulting display summarizes the network parameters for the currently running executor node.

Executor Characteristics as of 24-FEB-1989 15:21:13

Executor Node	= 2.975
Driver Version Number	= 3.0.0
State	= ON
Executor Identification	= DECNET-DOS V3.0
Executor Version	= 4.2.0
Node Name	= ROCKY
Node Address	= 2.975
Incoming Timer	= 45
Outgoing Timer	= 60
Confidence Timer	= 15
Incoming Proxy	= DISABLED
Outgoing Proxy	= ENABLED
Active Links	= 1
NSP Version	= 4.0.0
Maximum links	= 16
Delay Factor	= 32
Delay Weight	= 2
Inactivity Timer	= 30
Retransmit Factor	= 12
Routing Version	= 2.0.0


```

Routing Type           = Endnode IV
Maximum Buffers        = 24
Buffer Size            = 1498
Segment Buffer Size    = 557
Receive Password       = (password not set)
Transmit Password      = (password not set)
Database Path          = C:\DECNET\
Transmit pipe Quota    = 6
Receive pipe Quota     = 6
PC Type                = IBM PCAT
Nak Quota              = 0
Autoboot               = ENABLED
NCP>

```

4.1.8 Displaying Remote Node Information

To determine if you defined a specific node with the **DEFINE NODE** command, type **LIST NODE** followed by the node name. For example, if the node you want to check is **WALKER**, enter the following command:

```
NCP>LIST NODE WALKER Return
```

NCP displays the node name, node address, access control information, and the number of current active logical links for node **WALKER** from the local node.

```
Known Permanent Nodes as of 4-Feb-1989 16:36:30
```

```
Executor node           = 2.975 (ROCKY)
```

```
State                  On
Executor Identification DECnet-DOS V3.0
```

Node Address	Node Name	Active Links	LAT/ MS-NET	Account Information
2.379	WALKER	0		/SMITH/...
4.64	BLD1	1		

```
NCP>
```

If you want information only for nodes with active logical links, use the **SHOW ACTIVE NODES** command. For example:

```
NCP>SHOW ACTIVE NODES Return
```

This command displays the node name and address, the access control information, and the number of current active links from the local node to the adjacent node and any other nodes with active logical links to your node. The executor node is always displayed.

Known Volatile Nodes as of 4-Feb-1989 16:36:30

Node Address	Node Name	Active Links	LAT/ MS-NET	Account Information
4.64	BLD1	1		/SMITH/...

NCP>

4.1.9 Displaying Line Information

In the context of DECnet-DOS, a line is the physical line connecting the local node with the adjacent node. There are two possible line states:

- **ON**—The line is ready for use.
- **OFF**—The line's database and parameters are present, but the line is not available for any DECnet network activity.

There are three substates. They can be observed only when the line state is ON. The substates are:

- **SYNCHRONIZING**—The line is being initialized
- **RUNNING**—The line is in normal running state
- **SERVICE**—The line is reserved for loopback testing

To display status information about the physical line's characteristics, enter the following command:

NCP>SHOW LINE CHARACTERISTICS Return

The resulting display shows the current characteristics for the line.

Line Characteristics as of 24-FEB-1989 15:36:13

Line = ETHER-1

Line State	= ON
Line Substate	= Running
Device ID	= LPC-1
Receive Buffers	= 8
Protocol Type	= Ethernet
Hardware Address	= 08-00-2B-09-9C-89
Station Address	= AA-00-04-00-E8-13

NCP>

4.1.10 Displaying Circuit Information

In the context of DECnet-DOS, a circuit is the communications path established between the local node and a remote node with which you are communicating. There are two possible circuit states:

- **ON**—The circuit is ready for use.
- **OFF**—The circuit's database and parameters are present, but the circuit is not available for any DECnet network activity.

There are three circuit substates that further describe the circuit when it is ON. The substates are:

- **SYNCHRONIZING**—The circuit is being initialized.
- **STARTING**—The circuit is initialized.
- **RUNNING**—The circuit is active.

To display status information about the circuit's characteristics, enter the following command:

```
NCP>SHOW CIRCUIT CHARACTERISTICS Return
```

The resulting display shows the current characteristics for the circuit.

```
Circuit Characteristics as of 24-FEB-1989 15:39:44
```

```
Circuit = Ether-1
```

Circuit State	= ON
Circuit Substate	= Running
Service	= Enabled
Designated Router	= 4.378 ()
Block Size	= 1498
Hello Timer	= 30
Listen Timer	= 90
Verification	= 0
User	= DECnet
Owner	= DECnet
Line ID	= Ether-1
Protocol type	= Ethernet
Multicast Listener State	= Enabled

```
NCP>
```

4.1.11 Monitoring Events on Your Local Node

NCP provides several commands that allow you to monitor the activity of nodes on your network. For example, the MONITOR LOGGING command displays network events as they are being logged. An **event** is an occurrence that has potential significance in the operation and/or maintenance of a network. **Logging** is the process that generates a record of the event. Capturing this information in a file can be useful to you if network problems occur. The READ LOG command displays the contents of the event-logging buffer.

For DECnet-DOS systems, event-logging is always turned on, and all events are displayed. (You cannot select only a few events to be displayed.) After the events are displayed or redirected to a file, they are discarded.

If an event is lost, NCP displays an error message before displaying the rest of the logged events. For example:

```
Event type 0.0 Event records lost
```

For each event, the following information can be displayed:

```
Event type nn.nn name-of-event-type  
Occurred dd-mmm-yyyy hh:mm:ss  
other-event-dependent-text
```

See Chapter 6 for more information on the MONITOR LOGGING command.

4.2 Monitoring Other Nodes on the DECnet Network

The Network Management Listener (NML) responds to network management queries from remote DECnet systems. These systems must be running NCP or other network management tools that use the NICE V4.0 protocol.

NML enables other nodes on the network to query your node for network information. Similarly, you can query other nodes on your DECnet network for network information, if the nodes are running NML.

NML runs in the background so that it does not interfere with normal use of your personal computer. As a background task, it is limited to examining the currently running system (the volatile databases, whose information is displayed by using the SHOW commands). NML cannot access any information residing in the file system (the permanent database whose information is displayed by using the LIST commands).

4.3 NCP Commands Supported by DECnet-DOS NML

The following commands are supported from remote DECnet nodes to NML running on your personal computer:

NCP TELL *node-id* SHOW EXECUTOR $\left[\begin{array}{l} \text{SUMMARY} \\ \text{STATUS} \\ \text{CHARACTERISTICS} \\ \text{COUNTERS} \end{array} \right]$

NCP TELL *node-id* SHOW LINE $\left\{ \begin{array}{l} \text{ETHER-1} \\ \text{ASYNC-1} \end{array} \right\} \left[\begin{array}{l} \text{SUMMARY} \\ \text{STATUS} \\ \text{CHARACTERISTICS} \\ \text{COUNTERS} \end{array} \right]$

NCP TELL *node-id* SHOW KNOWN LINES $\left[\begin{array}{l} \text{SUMMARY} \\ \text{STATUS} \\ \text{CHARACTERISTICS} \\ \text{COUNTERS} \end{array} \right]$

NCP TELL *node-id* SHOW CIRCUIT $\left\{ \begin{array}{l} \text{ETHER-1} \\ \text{ASYNC-1} \end{array} \right\} \left[\begin{array}{l} \text{SUMMARY} \\ \text{STATUS} \\ \text{CHARACTER} \\ \text{COUNTERS} \end{array} \right]$

NCP TELL *node-id* SHOW KNOWN CIRCUITS $\left[\begin{array}{l} \text{SUMMARY} \\ \text{STATUS} \\ \text{CHARACTERISTICS} \\ \text{COUNTERS} \end{array} \right]$

NCP TELL *node-id* ZERO EXECUTOR COUNTERS

NCP TELL *node-id* ZERO LINE $\left\{ \begin{array}{l} \text{ETHER-1} \\ \text{ASYNC-1} \end{array} \right\}$ COUNTERS

NCP TELL *node-id* ZERO CIRCUIT $\left\{ \begin{array}{l} \text{ETHER-1} \\ \text{ASYNC-1} \end{array} \right\}$ COUNTERS

Testing the Network

This chapter provides an overview of the NCP commands that you can use to diagnose problems connecting your personal computer to the network. These commands allow you to display information about your node and the way it is interacting with the network. You can also display information about other nodes in the network.

You do not need special technical knowledge to use these commands, but it is helpful to know the **topology** of your network. The topology indicates which nodes are in the network, where they are located, and how they are connected. A basic understanding of network testing concepts (such as loopback) is also helpful.

This chapter is divided into two sections:

- **Overview of the NCP test commands**—describes the NCP test commands and lists the functions you can perform with them.
- **Testing the network**—describes the sequence of operations performed by the loopback commands as they test the network components (such as hardware, loopback connectors, or cables).

5.1 Overview of the NCP Test Commands

NCP provides a way for you to interact with the network. This interaction includes setting network parameters and displaying information about your node and the network. In addition, NCP provides a way to troubleshoot your network by issuing loopback tests from your node and allowing other nodes to send loopback tests to your node.

Because problems can occur in any of the network components, NCP commands let you test the operation of each component. The tests check the operation of your local node, the connection to the remote node, and the communication hardware between them.

By starting with your local node and progressing outward to each of the components in your network, you can determine which specific component is not operating properly. If all components appear to be working properly, you should contact your network manager or the manager of the node you are trying to reach. Have the manager perform further tests to determine the network problem.

The following commands let you display network information and allow other nodes to send loopback tests to your node:

- **LOOP** checks the operation of your local node, the connection to the remote node, and the communication hardware that connects them. This hardware includes local and remote modems.
- **MONITOR LOGGING** continuously displays event-logging information on the terminal screen.
- **READ LOG** displays the contents of the event-logging buffer.
- **SHOW** displays statistics about node, line, counter, and circuit characteristics.
- **TELL** instructs a remote DECnet™ node to display information about its lines, or circuits.
- **ZERO** resets line, circuit, or executor counters.

5.2 Testing the Network with Loopback Messages

You can use loop messages to test the following parts of your network:

- The local node
- The communications controller
- The Ethernet interface to the communications controller
- A remote node at the circuit level (only for Ethernet)
- A remote node at the application level

The loop messages are sent through the communications controller. DECnet-DOS™ V3.0 supports Ethernet connections using the integrated Ethernet communications controller, or asynchronous network connections using the integrated asynchronous serial port.

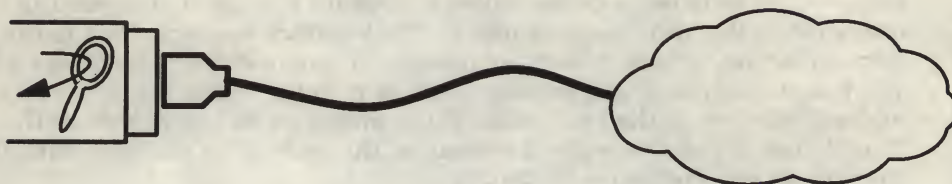
Some of the tests use a device called a loopback connector to check the operation of the various components. The loopback connector is a hardware device that you attach to each component to isolate it from the others while you test it. Loopback test messages are sent to the connector or network and echoed back to the local node. If the messages do not return, or if they do not match the original messages, the problem is probably with the component you are currently testing.

Loop tests require that the line and circuit states are ON before you run them. When you start the test, NCP displays a message indicating that the line or circuit is turned on. NCP turns lines and circuits ON before running the loop test.

The MIRROR command is used on the local node when a loopback test is performed from a remote node to the local node. MIRROR allows test messages from a remote node to be echoed back to that remote node.

The following figures show how you can progressively check the operation of your local node, your transceiver cable, your local modem, and your remote modem. These examples apply only to DDCMP configurations.

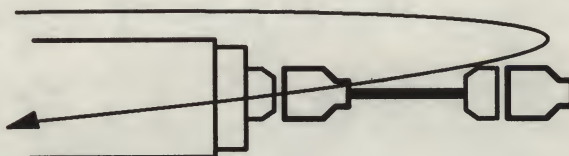
Figure 5-1: Loop Executor



LKG-3211-891

You can test the network software on your personal computer using the **LOOP EXECUTOR** command. You send the loopback messages through the network and routing software to yourself as shown in Figure 5-1. Your network software receives them and reports the receipt. If this test does not complete successfully, reinstall the network software.

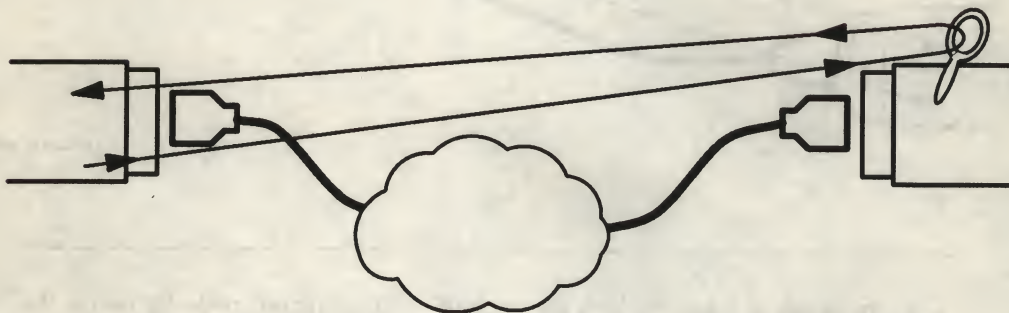
Figure 5-2: Loop Line Controller



LKG-3212-891

Figure 5-2 shows how you can test the network controller board on your personal computer by using the **LOOP LINE CONTROLLER** command. You must attach the loopback connector, included in your DECnet-DOS software kit, to run this test. The message is sent out through the network controller and is turned around at the loopback connector. If this test does not complete successfully, check the hardware controller on your personal computer.

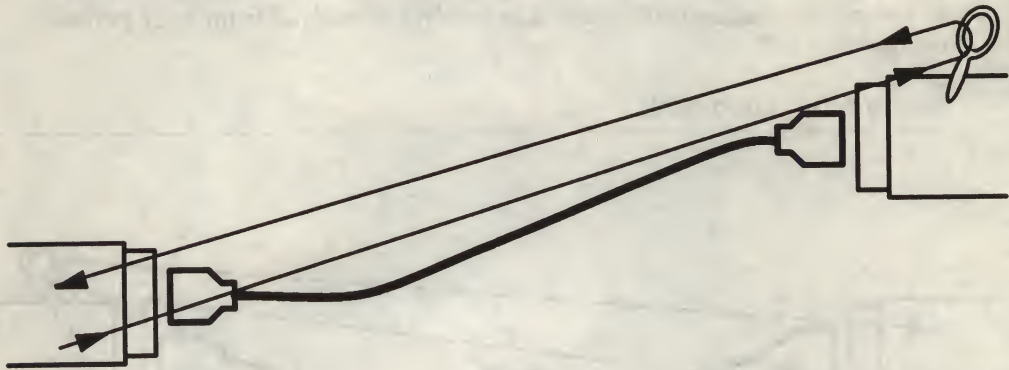
Figure 5-3: Loop Node



LKG-3214-891

You can check the connection to a node on your network by issuing the **LOOP NODE** command. With this command, you can send test messages through the network software, the network controller, and over the line to another node. A Loopback Mirror should be running on the node to echo the test message back to your node, as shown in Figure 5-3. If this test does not complete successfully, it indicates a problem with the connection to this remote node.

Figure 5-4: Loop Circuit



You can check the logical link connection to an adjacent node by using the **LOOP CIRCUIT** command. With this command, you can send test messages to a remote node through the network and routing software, the network controller, and over the circuit to an adjacent node. The loopback mirror on the adjacent node echoes the message back to your local node using the same logical link, as shown in Figure 5-4. If this test does not complete successfully, it indicates a problem along the link to the adjacent node.

5.2.1 Loopback Counters

When you run a loopback test, network counters record events that occur while the test is running. The counters record the error and traffic information on the network, including events for your local node, line, or circuit. You can zero these counters prior to running the loop test by using the **ZERO CIRCUIT**, **ZERO LINE**, and **ZERO EXECUTOR** commands. When you run the test, the errors and traffic that are recorded reflect the most recent network activity. To display the counters after you run the test, use one of the following commands:

- **SHOW CIRCUIT COUNTERS**
- **SHOW LINE COUNTERS**

- **SHOW EXECUTOR COUNTERS**

The loop tests are performed using the following information:

- The length of the test message
- The number of times to send the test message
- The type of format for the test message
- The node that will receive the test message and return it

Refer to Chapter 6 for more information on LOOP commands.

THE HISTORY OF THE

REIGN OF KING CHARLES THE FIRST

IN THE YEAR 1649

BY JOHN BURNET

OF THE UNIVERSITY OF OXFORD

IN TWO VOLUMES

LONDON: Printed by J. Streater, at the Sign of the Gun, in St. Dunstons Church-yard, 1704.

NCP Commands for DECnet-DOS

This chapter provides detailed descriptions of all the NCP commands that you can use for setting up, maintaining, updating, and monitoring your node in a DECnet-DOS™ network.

6.1 NCP Commands

The NCP commands are presented in alphabetical order in the following sections. Each section explains how to use the command and provides command examples and sample system responses to the command.

Table 6-1 summarizes all the NCP commands and their definitions.

Table 6-1: NCP Command Summary

Command	Function
CLEAR EXECUTOR	Removes executor node parameters stored for the currently running system.
CLEAR LOCAL-ADAPTER- NAME	Clears a local adapter name from the volatile database.
CLEAR REMOTE-ADAPTER- NAME	Clears a remote adapter name from the volatile database DECRET.DAT.
COPY KNOWN NODES	Copies the node database to a specified file. Access information is not copied.

Table 6-1 (Cont.): NCP Command Summary

Command	Function
COPY NODE	Copies information about a single node name from a specified remote node into the node database.
DEFINE ACCESS	Defines incoming access information for the designated user in the incoming access database (DECACC.DAT). Access information includes a user ID, a password, and the type of access.
DEFINE CIRCUIT	Defines the circuit characteristics that take effect when you restart DECnet-DOS. TM
DEFINE EXECUTOR	Defines executor node characteristics that take effect when you restart DECnet.
DEFINE LINE	Defines line characteristics that take effect when you restart DECnet.
DEFINE NODE	Assigns a node name and optional access control information to a unique node address.
DEFINE OBJECT	Defines object information for the Spawner database (DECOBJ.DAT).
DEFINE REMOTE-ADAPTER- NAME	Defines a remote adapter name to be used by NETBIOS. The adapter name includes a node name and an object associated with that node.
EXIT	Exits from NCP.
HELP	Displays a summary of NCP commands on your screen.
LIST ACCESS	Displays entries in the incoming access database (DECACC.DAT).
LIST CIRCUIT	Displays permanent information about the circuit from the DECnet database (DECPARM.DAT).
LIST EXECUTOR	Displays permanent information from the DECnet database for the executor node.
LIST KNOWN	Displays permanent information about all occurrences of an entity from the DECnet database (DECPARM.DAT, DECNODE.DAT, DECOBJ.DAT, DECREM.DAT, and DECACC.DAT).
LIST LINE	Displays permanent information about the line from the DECnet database (DECPARM.DAT).

Table 6-1 (Cont.): NCP Command Summary

Command	Function
LIST NODE	Displays permanent information about a remote node that is contained in the node database (DECNODE.DAT).
LIST OBJECT	Displays information about an object in the Spawner database (DECOBJ.DAT).
LIST REMOTE-ADAPTER- NAME	Displays information about the remote adapter name in the remote adapter name database (DECREM.DAT).
LOOP CIRCUIT	Runs loopback tests to a specific Ethernet address, a multicast Ethernet address, or an adjacent node on a DDCMP line.
LOOP EXECUTOR	Runs a loopback test within the local node to a local loopback mirror.
LOOP LINE	Runs a loopback test within the local node to the Ethernet controller. This test requires the use of a loopback connector.
LOOP NODE	Runs a loopback test to a remote node that has a loopback mirror.
MONITOR LOGGING	Displays event-logging information on the screen.
PURGE ACCESS	Deletes incoming access information for the specified user from the incoming access database (DECACC.DAT).
PURGE EXECUTOR	Removes specified executor node parameters from the DECnet database (DECPARM.DAT).
PURGE NODE	Deletes information in the local node database for the specified node.
PURGE OBJECT	Deletes information in the Spawner database (DECOBJ.DAT) using either the object name or object number.
PURGE REMOTE-ADAPTER- NAME	Deletes an entry from the remote adapter name database (DECREM.DAT).
READ LOG	Displays the contents of the event-logging buffer either on the screen or in a redirected file.

Table 6-1 (Cont.): NCP Command Summary

Command	Function
SET CIRCUIT	Sets or modifies current circuit parameters.
SET ECHO	Determines whether the contents of a command file or a redirected file are displayed on your screen as the file is executed.
SET EXECUTOR	Sets or modifies the executor node's current parameters.
SET LINE	Sets or modifies the line's current parameters.
SET LINK	Terminates the logical link connections associated with a specific socket.
SET LOCAL-ADAPTER-NAME	Enters a local adapter name in the volatile database.
SET PAUSE	Causes a long NCP display to pause at the end of a screen. This prevents the entire display from scrolling by. This command also issues a prompt so that you can view the rest of the display.
SET REMOTE-ADAPTER-NAME	Enters a remote adapter name in the volatile database.
SHOW ACTIVE LINKS	Displays information to your local node about current logical links.
SHOW ADJACENT NODES	Displays information about the node that is physically adjacent to your local node.
SHOW CIRCUIT	Displays current circuit information.
SHOW EXECUTOR	Displays current executor node information.
SHOW KNOWN LINKS	Displays information about all logical links to the local node.
SHOW KNOWN NODES	Displays information about nodes that have been accessed by DECnet-DOS since the last time you stopped DECnet. To see permanent node information, use the LIST KNOWN NODES command.
SHOW LAT	Displays counters for LAT (applies if the LAT driver is installed).

Table 6-1 (Cont.): NCP Command Summary

Command	Function
SHOW LINE	Displays communication line information.
SHOW LOCAL-ADAPTER- NAME	Displays information about local adapter names.
SHOW REMOTE-ADAPTER- NAME	Displays information about remote adapter names.
TELL <i>node-id</i>	Instructs a remote node to display information about its executor, lines, or circuits.
ZERO CIRCUIT	Zeroes the counters for the circuit.
ZERO EXECUTOR	Zeroes the counters for the executor node.
ZERO LAT	Zeroes counters for LAT (applies if the LAT driver is installed).
ZERO LINE	Zeroes the counters for the line.

CLEAR

CLEAR

The CLEAR command removes specified parameters from the local volatile database for the running system. The KNOWN keyword clears all information about the parameter from the volatile database. To remove these parameters from the permanent database, use the PURGE command.

- **LOCAL-ADAPTER-NAME**—Identifies the network application written for the NETBIOS emulator interface.
- **REMOTE-ADAPTER-NAME**—Identifies a remote network application written for the NETBIOS emulator interface.

CLEAR EXECUTOR

The **CLEAR EXECUTOR** command removes specified executor node parameters from the local volatile database for the running system. These parameters include a **RECEIVE PASSWORD** and a **TRANSMIT PASSWORD**. To remove these parameters from the permanent database, use the **PURGE EXECUTOR** command.

If **RECEIVE** and **TRANSMIT** passwords have been enabled on the executor node and the adjacent node, the passwords are automatically exchanged when either node turns on the communication line to the other node. The network coordinator defines these passwords for **DDCMP** use. The exchange of passwords also occurs when the communication line is restarted from either node because of errors.

Format

CLEAR EXECUTOR { **RECEIVE PASSWORD**
 TRANSMIT PASSWORD }

where

**TRANSMIT
PASSWORD** removes the password the executor node must receive from the adjacent node in order to exchange messages with the adjacent node. The password is a character string consisting of eight alphanumeric characters.

**RECEIVE
PASSWORD** removes the password the executor node must transmit to the adjacent node in order to exchange messages with the adjacent node. The password is a character string consisting of eight alphanumeric characters.

NOTE

These passwords are used for asynchronous **DDCMP** connections.

CLEAR EXECUTOR

Example

The following command removes the password that the executor node must transmit to the adjacent node when the executor node first begins communications. Until the network coordinator defines a new transmit password, the executor node cannot open the communication line between itself and the adjacent node.

```
NCP> CLEAR EXECUTOR TRANSMIT PASSWORD Return
```

CLEAR LOCAL-ADAPTER-NAME

This command clears a specific local adapter name from the volatile database. The local adapter name identifies a network application written for the NETBIOS emulator interface. The network application resides on your personal computer.

The CLEAR KNOWN LOCAL-ADAPTER-NAMES command clears all the local NETBIOS names from the volatile database.

The CLEAR command for adapter name information is case sensitive. This means that you must specify the exact name to clear by enclosing the name in quotation marks. If you have two local adapter names that use the same letters but are in different cases, such as "BEST" and "best", you must be very specific when using the CLEAR LOCAL-ADAPTER-NAME command. If you do not use quotation marks with the CLEAR command, NCP will remove the name that is stored in all capital letters.

Format

CLEAR LOCAL-ADAPTER-NAME *name* [BYTE16 *nn*]

or

CLEAR KNOWN LOCAL-ADAPTER-NAMES

where

name is a 1- to 16-character ASCII printable string that specifies the name of the local application to remove from the volatile database.

nn specifies a value for byte 16. This value can be any decimal value from 0 to 255.

CLEAR LOCAL-ADAPTER-NAME

Example

The following example removes the entry, "Charlie", from the volatile database.

```
NCP> CLEAR LOCAL-ADAPTER-NAME "Charlie" Return
```

CLEAR REMOTE-ADAPTER-NAME

This command clears a specific remote adapter name from the volatile database. The remote adapter name identifies a network application written for the NETBIOS emulator interface. The network application resides on a remote personal computer.

The CLEAR KNOWN REMOTE-ADAPTER-NAMES command clears all the remote NETBIOS names from the volatile database.

The CLEAR command for adapter name information is case sensitive. This means that you must specify the exact name to clear by enclosing the name in quotation marks. If you have two local adapter names that use the same letters but are in different cases, such as "ANCHOR" and "anchor", you must be very specific when using the CLEAR REMOTE-ADAPTER-NAME command. If you do not use quotation marks with the CLEAR command, NCP will remove the name that is stored in all capital letters.

Format

CLEAR REMOTE-ADAPTER-NAME *name*

or

CLEAR KNOWN REMOTE-ADAPTER-NAMES

where

name is a 1- to 16-character alphabetic string that specifies the name of the remote application to remove from the volatile database.

Example

The following example removes the entry, "Hoosier", from the volatile database.

NCP> CLEAR REMOTE-ADAPTER-NAME "Hoosier" Return

COPY NODE

COPY NODE

The **COPY NODE** *node-id* **FROM** command copies information about a single node name (or a group of nodes within a single area) from a specified remote node into the node database (DECNODE.DAT). Use this command to add information from a particular node to your database. Use this command also to copy information from a node with a large node database if you want to copy information about nodes in a particular area.

The **COPY KNOWN NODES FROM** command specifies a remote node from which to copy the complete database of node names and node addresses to your local node. This set of node names will not include any access information. If the database on the remote node is large (for example, in excess of 100 nodes), this can significantly slow down the access time to the remote node's database. You can interrupt the copying of node names at any time by pressing any key on the keyboard.

The **COPY KNOWN NODES TO** command copies the local node database to a file. If the database contains access information, that information is not copied.

Format

COPY NODE { *node-id*
 *area.** } **FROM** *remote-node-id*

or

COPY KNOWN NODES FROM *remote-node-id* [TO *file-id*]

where

node-id specifies the name or the address of the node for which to collect information.

remote-node-id specifies the name or the address of the remote node from which to collect information.

*area.**

specifies the particular node area for which you want to collect information. The asterisk (*) is a wildcard that indicates all the node information for the specified area. (The use of wildcards works if the remote system supports wildcarded node names for network management.)

TO file-id

identifies the file in which the output of the COPY NODE command is stored.

Examples

The following command copies information about the node BASBAL from the node STADIA.

```
NCP> COPY NODE BASBAL FROM STADIA Return
```

The following command copies the node database to the file NODEDATA.DAT.

```
NCP> COPY KNOWN NODES TO NODEDATA.DAT Return
```


DEFINE

DEFINE

The **DEFINE** command changes values of various parameters in the DECnet-DOS permanent databases. The following are parameters that you can change by using the **DEFINE** command:

- **ACCESS**—The incoming access information for a remote node that wants to access your local node.
- **CIRCUIT**—The characteristics of a connection to a remote node.
- **EXECUTOR**—The characteristics of your local node as an executor node.
- **LINE**—The characteristics of the physical connection to an adjacent node.
- **NODE**—The address and access information for a remote node to which your local node will connect.
- **OBJECT**—The programs that the Spawner can run when a request is made from a remote node.
- **REMOTE-ADAPTER-NAME**—The node on which a remote application resides.

When you restart your DECnet-DOS node, the software reads the **DECPARM.DAT** database to find the characteristics of the executor, circuit, and line, and creates a volatile database using this information. You will not notice the results of the **DEFINE** command on these parameters until you restart DECnet. For the executor, line, or circuit values to take effect immediately use the **SET** command.

No volatile database is created for nodes, access information, or object information. You can change these permanent databases dynamically using the **DEFINE** command. Changes are made directly to the corresponding permanent database on disk. The DECnet software reads this information on disk as it is needed (no restart is necessary).

DEFINE ACCESS

The **DEFINE ACCESS** command defines incoming access information for the designated user. The access information is stored for each user, and it includes the password and access type. The access type describes the type of privilege the user has to access information on your node. If no access type is defined for the user, the access for that user is **NONE**.

The information you define with **DEFINE ACCESS** is stored in the database **DECACC.DAT**. The **DECACC.DAT** file is used by the File Access Listener (FAL) and other DECnet-DOS utilities to limit access from remote systems.

For the characters representing the access information to be stored in uppercase letters, type the characters. For the characters to be stored as typed, place the string within quotation marks. For example:

abc is stored in the database as **ABC**

"Aab" is stored in the database as **Aab**

Format

DEFINE ACCESS USER *username* [**PASSWORD** *password*]

[**NONE**
READ
WRITE
ALL]

where

username is a 1- to 39-character alphabetic string that defines the user.

password is a 1- to 39-character alphabetic string that defines the user's password. The default is no password.

DEFINE ACCESS

access-type

describes the type of privilege the user has for accessing information on this node. There are four types of access:

- NONE—no access for this user.
- READ—provides read-only access.
- WRITE—provides write-only access.
- ALL—provides both read and write access.

Example

The following command defines access information for the user BARKER. In this case, the password is JUMPER and the access type is READ. This means that the user BARKER can read information on your node.

```
NCP> DEFINE ACCESS USER BARKER PASSWORD JUMPER READ Return
```

DEFINE CIRCUIT

The **DEFINE CIRCUIT** command defines permanent characteristics for the circuit. The value you specify does not take effect until the next time you boot your system or restart the network software. (For the value to take effect immediately but temporarily, use the **SET CIRCUIT** command.)

Format

```

DEFINE CIRCUIT {
    HELLO TIMER seconds
    MULTICAST-LISTENER { DISABLED
                        ENABLED }
    OWNER { DECNET
            DLX
            MOP }
    SERVICE { DISABLED
             ENABLED }
    STATE { ON
           OFF }
}
    
```

where

HELLO TIMER
seconds

specifies the frequency of routing hello messages sent to the adjacent node over the circuit. The valid range is 1 to 8191 seconds. The default is 30.

MULTICAST-LISTENER sets the circuit's multicast-listener state. The default is **ENABLED**.

DISABLED

specifies that the circuit will not respond to multicast requests. This applies to Ethernet communications.

ENABLED

specifies that the circuit will respond to multicast requests. This applies to Ethernet communications.

OWNER

specifies the circuit owner. The default is **DECNET**.

SERVICE

sets the circuit's service state. The default is **ENABLED**.

DEFINE CIRCUIT

DISABLED

specifies that the circuit is unavailable for service operations.

ENABLED

specifies that the circuit is available for special network activity such as loopback testing.

STATE

sets the line's operational state to ON or OFF. The default is ON.

ON

allows network traffic to flow over the circuit. This is the normal operational state.

OFF

allows no traffic to flow over the circuit. The circuit is unavailable for any network activity.

Example

The following command sets the circuit state to ON for the next time you restart DECnet.

```
NCP> DEFINE CIRCUIT STATE ON Return
```

DEFINE EXECUTOR

The **DEFINE EXECUTOR** command defines executor characteristics that become effective after you restart DECnet. (For the value to take effect immediately but temporarily, use the **SET EXECUTOR** command.)

Format

```
DEFINE EXECUTOR
ADDRESS node-address
CONFIDENCE TIMER number
DELAY FACTOR number
DELAY WEIGHT number
IDENTIFICATION id-string
INACTIVITY TIMER seconds
INCOMING TIMER seconds
MAXIMUM BUFFERS number
MAXIMUM LINKS number
NAK QUOTA number
NAME node-name
OUTGOING TIMER seconds
RECEIVE PASSWORD password
RECEIVE PIPE QUOTA number
RETRANSMIT FACTOR number
BUFFER SIZE number
SEGMENT BUFFER SIZE number
STATE [ON | OFF]
TRANSMIT PASSWORD password
TRANSMIT PIPE QUOTA number
AUTOBOOT [ENABLED | DISABLED]
```

where

ADDRESS <i>node-address</i>	specifies the address assigned to the executor node. The node address is a unique numeric identification for the node. It must include an area number and a node number.
CONFIDENCE TIMER <i>number</i>	an optional extra amount of time to wait before timing out a link. The value range for this timer is 1 to 32K. The default is 15.

DEFINE EXECUTOR

DELAY FACTOR
number

determines the amount of time to wait before a message is retransmitted to a node. The range is 16 to 4095. The default number is 32.

DELAY WEIGHT
number

is the weight to apply to a current round-trip delay to a remote node when updating the estimated round-trip delay to a node. The range is 1 to 255. The default is 1.

IDENTIFICATION
id-string

specifies a text identification string for the executor node. The string can consist of 1 to 32 alphanumeric characters. You must use a set of double quotation marks (" ") to delimit any string containing blanks or tabs. You cannot use a single set of quotation marks within the identification string.

You can use the IDENTIFICATION parameter to identify the executor node to other nodes in the network (for example, by including your name or department in the string).

INACTIVITY TIMER
seconds

is the maximum duration of inactivity on a logical link before the node checks to see if the logical link still works. If no activity occurs within the minimum number of seconds, the node generates traffic to test the link. The value range is 1 to 32K. The default is 30.

INCOMING TIMER
seconds

is the maximum duration between the time a connection request is received at the executor node and the time that it takes to accept or reject the request. If the request is not accepted or rejected by the user program within the specified number of seconds, the connection request is automatically rejected. If no value is set, there is no timer. The value range is 1 to 32K. The default is 45.

MAXIMUM BUFFERS
number

specifies the maximum number of network communication buffers available for use by DECnet-DOS. The valid range is 1 to 255. The actual number of buffers is restricted by available memory. This affects the memory usage. The default is 16.

MAXIMUM LINKS

number

specifies the maximum number of active logical links for the executor node. The valid range is 1 to 32. The default is 4.

NAK QUOTA *number*

is the number of out-of-order segments to receive before sending a negative acknowledgment (NAK) out on the logical link. This number must be less than or equal to the RECEIVE PIPE QUOTA. The value range for this quota is 0 to 4095. The default is 0 (or 6 for slower devices and processors).

NAME *node-name*

specifies the node name for the executor node. The node name is an identification string consisting of one to six alphanumeric characters including at least one alphabetic character.

OUTGOING TIMER

seconds

is the duration between the time that the executor node requests a connection and the time that it takes for the request to be acknowledged at the remote node. The value range is 1 to 65535. The default is 60.

RECEIVE PASSWORD

password

is the password the local node must receive from the adjacent node in order to exchange messages with the adjacent node. The password is a character string consisting of eight alphanumeric characters.

RECEIVE PIPE QUOTA

number

is the number of segments the network will receive for a link by flow control. The valid range is 1 to 65535. The default is 6.

RETRANSMIT FACTOR

number

is the maximum number of times that the executor node will restart the retransmission timer when it expires. If the number is exceeded, the logical link is disconnected. The range is 1 to 65535. The default is 12.

BUFFER SIZE *number*

allows you to define the maximum buffersize for Wide Area Network messages. These buffers are for messages within the local area network. The maximum value is 1498 bytes. This is also the default value.

DEFINE EXECUTOR

SEGMENT BUFFER SIZE *number*

specifies the size of the segment buffers in bytes. These buffers are used for transmitting DECnet messages from the local area network. The size is a decimal integer in the range of 1 to 16383 for DDCMP communications and 1 to 1466 for Ethernet. The default is the user buffer size + 17.

STATE ON

allows logical links to be established to the executor node. The executor's state is OFF if the DECnet network process fails to find the DECnet database file, DECPARM.DAT. When the state is set to OFF, the system will restart with the executor state OFF.

TRANSMIT PASSWORD *password*

is the password the local node must transmit to the adjacent node in order to exchange messages with the adjacent node. The password is a character string consisting of eight alphanumeric characters.

TRANSMIT PIPE QUOTA *number*

is the number of segments the driver will buffer into the network before it blocks the user on a transmission. The valid range is 1 to 65535. The default is 6.

AUTOBOOT

specifies that the node can be automatically booted from a remote node. This parameter is used by PCSA. The default is ENABLED.

Examples

The following command defines the executor address as 2.975.

```
NCP> DEFINE EXECUTOR ADDRESS 2.975 Return
```

The following command defines the executor address as 2.975 and assigns ROCKY as the executor name.

```
NCP> DEFINE EXECUTOR ADDRESS 2.975 NAME ROCKY Return
```

The following command defines the maximum number of logical links for the executor node as 6.

```
NCP> DEFINE EXECUTOR MAXIMUM LINKS 6 Return
```

DEFINE LINE

The **DEFINE LINE** command defines the state of the line at network restart. It becomes effective with the next network restart. (For the value to take effect immediately but temporarily, use the **SET LINE** command.)

Format

```
DEFINE LINE [line-id] { DEVICE [device-id]
                           MODEM { NULL
                                   FULL
                           }
                           RECEIVE BUFFERS number
                           SPEED baud-rate
                           STATE { ON
                                   OFF
                           }
```

LINE [*line-id*] specifies either **ASYNC-1** or **ETHER-1** as the line for which parameters are to be created or modified. Because there is one line for each configuration, the *line-id* is optional.

DEVICE [*device-id*] specifies the device for controlling the asynchronous DDCMP line (**COM-1**, **COM-2**). The line's circuit must be in the **OFF** state.

MODEM specifies the type of modem control to be used. Modem control is used with asynchronous DDCMP connections.

NULL specifies that modem control signals will not be checked.

FULL specifies that modem control signals are checked. The line is shut down if the modem loses its connection. This is the default.

RECEIVE BUFFERS *number* specifies the number of buffers for receiving information. Increasing this number affects network response time and memory usage (for use in Ethernet communications). The valid range is 1 to 1024. The installation procedure sets the default based on the configuration of your node.

DEFINE LINE

SPEED *baud-rate*

sets the DDCMP line speed to one of the following:

50	150	1200	3600
75	200	1800	4800
110	300	2000	7200
134	600	2400	9600

The default is 9600. For high performance systems, a speed of 19200 is available.

STATE

defines the line's restart state to ON or OFF. When the line state is ON, the line is available for network use. When the line state is OFF, the line's database and parameters are present, but the line is not enabled for any type of network activity. For asynchronous communications, the port is available for nonDECnet use.

If there is no DECPARM.DAT file, the default line state for restart is OFF.

NOTE

Information about modem, line speed, and service state applies to asynchronous DDCMP communications.

Examples

The following command defines the line state as ON. When you restart DECnet, the line will come up in the ON state.

```
NCP> DEFINE LINE STATE ON Return
```

The following command defines the maximum number of receive buffers for Ethernet communication as 6.

```
NCP> DEFINE LINE RECEIVE BUFFERS 6 Return
```

DEFINE NODE

The **DEFINE NODE** command assigns a node name plus access control information to a unique node address. For the characters representing the access information to be stored in uppercase letters, type the characters. For the characters to be stored as typed, place the string within quotation marks. For example:

abc is stored in the database as **ABC**

"Aab" is stored in the database as **Aab**

Format

DEFINE NODE *node-address* **NAME** *node-name*

<i>access-info</i> LAT-HOST MS-NET
--

where

NODE

node-address

specifies the address of the node to be defined. The node address is a unique numeric identification for the node. It must include an area number and a node number.

NAME

node-name

specifies the name to be associated with the node address. The node name is an identification string consisting of one to six alphanumeric characters including at least one alphabetic character.

DEFINE NODE

access-info

specifies access control information for the remote node. This information uses the following format:

USER *user-name* [PASSWORD *password*] [ACCOUNT *account*]

where

user-name is a character string of 1 to 39 alphanumeric characters that identifies the user at the remote node.

password is a character string of 1 to 39 alphanumeric characters that might be required to access files or programs on the remote node.

account is a character string of 1 to 39 alphanumeric characters.

MS-NET

indicates that the node is an MS-NET server. If you have PCSA installed, this command will cause an entry in the NETBIOS server volatile database. (Also refer to the PURGE NODE command.)

LAT-HOST

indicates that the node is a LAT host.

Examples

The following command assigns the node name RM2 to the node at address 2.65. It also specifies the user-name as MAGIC and the password as WAND.

```
NCP> DEFINE NODE 2.65 NAME RM2 USER MAGIC PASSWORD WAND Return
```

To add access control information or a node designation to a node that you have already defined in the database, you do not have to type both the node name and node address to add the information. You can use either the node's name or its address. For instance, if you have already defined node BORIS with its address as 1.2, you can use either the name or the number in the following example.

```
NCP> DEFINE NODE BORIS USER BENCH Return
```

This command assigns access information for the user BENCH to the previously defined node BORIS using the node name.

DEFINE OBJECT

The **DEFINE OBJECT** command defines object information for the Job Spawner. The information is stored in the object database, **DECOBJ.DAT**. The Spawner uses the database information to start a program or application whenever it receives a connect request from other nodes in the network. For more information about using the Job Spawner, refer to the *DECnet-DOS User's Guide*.

For the object information to be stored in uppercase letters, type the characters. If you do not want the characters to be stored as uppercase, place the string within quotation marks. For example:

abc is stored in the database as **ABC**

"Aab" is stored in the database as **Aab**

Format

DEFINE OBJECT *object-name* [**NUMBER** *number*] **FILE** *file-name* [**ARGUMENTS** *xxx*]

where

<i>object-name</i>	is the name of the DECnet object. The object name can have a maximum length of 16 characters.
<i>number</i>	is the number of the DECnet object. The range for this number is 0 to 255.
<i>file-name</i>	is the name of a command file or program to be started by the Spawner. Command file names must end with the .CMD extension. (The file name can include a path specification.)

DEFINE OBJECT

xxx

are the command-line arguments for the program specified with FILE. If there are spaces embedded within the string, be sure to enclose the string in quotation marks.

DECnet-DOS reserves the numbers 1-127 for DECnet objects. Each number is predefined for a network program. The object numbers 128-255 are available for user-written or user-supplied applications. The following table lists the DECnet-DOS predefined object numbers:

Object #	Object Name	Object File Name
0	Any object	Any object file name
17	FAL	FAL.EXE
63	DTR	DTR.EXE
128-255	Any user-written/ provided network object	Any user-defined name

The object number zero has a special meaning. An unlimited number of objects can be defined as object zero. When DECnet receives a connection request of object number zero, it makes the connection based on name.

NOTE

The DECnet-DOS installation procedure defines some objects for you during the installation process. The installation procedure lets you decide what objects to install. You can use NCP commands to change or delete these objects or to define new objects.

Examples

The following command defines the File Access Listener (FAL) as object number 17.

```
NCP> DEFINE OBJECT FAL NUMBER 17 FILE FAL Return
```

The following command defines a command file (MYCOM.CMD) as object number 0. When a connection for MYCOM.CMD is requested, DECnet will match the connection request object name with the object MYCOM.CMD in

DEFINE OBJECT

the DECOBJ.DAT database. The connection will not be based on the object number 0.

```
NCP> DEFINE OBJECT MYCOM FILE NUMBER 0 MYCOM.CMD ARGUMENTS a b Return
```

The following command defines a command file (TST.CMD) as object number 128. The arguments for this file are stored in the database exactly as indicated in the quoted string.

```
NCP> DEFINE OBJECT TST NUMBER 128 FILE TST.CMD ARGUMENTS "A B" Return
```


DEFINE REMOTE-ADAPTER-NAME

DEFINE REMOTE-ADAPTER-NAME

The DEFINE REMOTE-ADAPTER-NAME command associates a remote adapter name with a remote node and object. The name is stored in the remote adapter database, DECREM.DAT. The adapter name includes a node name and a particular object associated with that node name.

A remote adapter is a network application written for the NETBIOS emulator interface. The application resides on a remote node.

For the value to take effect immediately but temporarily, use the SET REMOTE-ADAPTER-NAME command.

For the user information to be stored in uppercase letters, type the characters. If you do not want the characters to be stored as uppercase, place the string within quotation marks. For example:

abc is stored in the database as **ABC**

"Aab" is stored in the database as **Aab**

Format

DEFINE REMOTE-ADAPTER-NAME *name* NODE *node-id* OBJECT *number*

USER <i>user-id</i>
PASSWORD <i>password</i>
ACCOUNT <i>account</i>

where

name is a 1- to 16-character alphabetic string that specifies the name to assign to the remote application. For the characters representing the name to be stored in uppercase letters, type the characters. If you do not want the characters to be stored as uppercase, place the string in double quotation marks.

DEFINE REMOTE-ADAPTER-NAME

<i>node-id</i>	identifies the node that is associated with this remote adapter. The node you use must be one that is already defined in the node database, DECNODE.DAT. You must identify a node for each remote adapter name that you define. The node-id can be either the node's name or its address. A node name is a 1- to 6-character alphanumeric string that identifies a specific node. A node address is a numeric character string consisting of an area number and a specific node number. The node address also serves as a unique identification of the node in the network.
<i>number</i>	is the number of the DECnet object. The range for this number is 0 to 255. The default is 0.
<i>user-id</i>	is the user information you use for accessing the loopback mirror on the remote node. A user ID consists of 1 to 39 alphabetic characters.
<i>password</i>	is a unique character string that you use for accessing the loopback mirror on the remote node. A user's password consists of 1 to 39 alphanumeric characters (some systems restrict password lengths to 8 characters).
<i>account</i>	is alphanumeric information you need to access the adjacent node's loopback mirror. The information can consist of 1 to 39 alphanumeric characters.

Example

The following command causes the remote adapter name WINSTON to be stored in the DECRET.DAT database. Whenever a connection request with the number 42 is made, the Spawner will access node WINSTON for the network application.

```
NCP> DEFINE REMOTE-ADAPTER-NAME WINSTON NODE 2.67 OBJECT 42 Return
```


EXIT

EXIT

The **EXIT** command lets you exit from NCP. You can also use **CTRL/Z** **Return** to exit from NCP.

Format

EXIT

Example

The following command causes you to exit from NCP and returns you to the prompt or system you were using previously.

NCP> **EXIT** **Return**

HELP

The **HELP** command displays information about NCP commands on your screen.

Format

HELP [*command-verb* [*entity*]]

where

command-verb specifies the NCP command for which you need information.

entity specifies that you need information about the specific command as it relates to this component.

Example

The following command causes **HELP** information to be displayed for the **LIST NODE** command.

NCP> **HELP LIST NODE** Return

The system responds with:

Display information about a remote node contained in the permanent database.

LIST NODE [<i>node-id</i>]	[TO <i>file-name</i>]
LIST NODE [<i>area.*</i>]	[TO <i>file-name</i>]
LIST NODE [<i>*.*</i>]	[TO <i>file-name</i>]

LIST

LIST

The **LIST** command displays the contents of various permanent databases. The following parameters can be listed:

- **ACCESS**—Displays the access information for a node or all known nodes.
- **CIRCUIT**—Displays the characteristics of the circuit.
- **EXECUTOR**—Displays the characteristics of the local node as an executor node. These are the characteristics the node possesses when you restart DECnet.
- **LINE**—Displays the characteristics of the line to an adjacent node.
- **NODE**—Displays the nodes known to your local node when you restart DECnet.
- **OBJECT**—Displays the objects known by the Spawner when you restart DECnet.
- **REMOTE-ADAPTER-NAME**—Displays the remote adapter names known by the Spawner at restart.

The contents of these databases are the default settings your node will take when you restart DECnet. The current characteristics of your node are stored in the volatile database. They can be displayed with the **SHOW** command.

LIST ACCESS

The LIST ACCESS command displays an entry in the incoming access database (DECACC.DAT) for a specific user. The access information is stored for each user, and it includes the password and access type. The access type describes the type of privilege the user has to access information on your node, such as READ or WRITE privileges.

If passwords are included as part of the incoming access information, they are not displayed when you issue the LIST ACCESS command.

The LIST KNOWN ACCESS command displays all entries contained in the incoming access database. You can also redirect this information to an output file for future reference.

Format

LIST ACCESS USER *user-name* [TO *file-id*]

or

LIST KNOWN ACCESS [TO *file-id*]

where

user-name is a 1 to 39 alphabetic character string that defines the user whose access information you want to display. For the username to appear in lowercase letters, be sure to enclose the string in quotation marks.

TO *file-id* specifies the name of an output file to which information is to be directed.

LIST ACCESS

Example

The following command displays the contents of the incoming database.

```
NCP> LIST KNOWN ACCESS Return
```

Default incoming access is NONE

Access Type	User/Password
ALL	GEORGE
ALL	PAUL/...
READ	RINGO

NCP>

LIST CIRCUIT

The LIST CIRCUIT command displays all the permanent parameters you can change using the DEFINE CIRCUIT command. You can either display the information on your screen or redirect it to an output file at the local node.

This command displays information that you can change. To view all the information about the circuit, use the SHOW CIRCUIT command.

Format

```
LIST CIRCUIT [circuit-id] [ CHARACTERISTICS  
                           STATUS  
                           SUMMARY ] [TO file-id]
```

where

CIRCUIT [<i>circuit-id</i>]	ASYN or ETHER-1 specifies the circuit for which information is to be displayed. Because there is one circuit to use in DECnet-DOS, the <i>circuit-id</i> is optional.
CHARACTERISTICS	displays parameters currently set for the executor, line, or circuit.
STATUS	displays the availability of the circuit for network activity.
SUMMARY	displays a summary of information for the circuit.
TO <i>file-id</i>	specifies the name of an output file to which information is to be directed.

Examples

The following command displays the permanent attributes of circuit ETHER-1.

```
NCP> LIST CIRCUIT ETHER-1 CHARACTERISTICS Return
```


LIST CIRCUIT

Circuit Characteristics as of 9-Mar-1989 9:28:12

```
Circuit state      = On
Service           = Enabled
Hello timer       = 30
Verification      = 0
Owner             = DECnet
Multicast listener state = Enabled
NCP>
```

The following command redirects the permanent attribute information about circuit ETHER-1 to an output file named CIR.DAT. To see the information in CIR.DAT you must exit NCP, by typing **Ctrl/Z** **Return**, and TYPE the file.

```
NCP> LIST CIRCUIT ETHER-1 SUMMARY TO CIR.DAT Return
```

```
C:\> TYPE CIR.DAT Return
```

Circuit Summary as of 9-Mar-1989 9:30:31

Circuit	State/Substate	Designated Router	Block Size
ETHER-1	On/Running		1498

NCP>

LIST EXECUTOR

The LIST EXECUTOR command displays permanent information about the local node. This command displays information that you can change. To view all the current information about the local node, use the SHOW EXECUTOR command.

Format

```
LIST EXECUTOR [ CHARACTERISTICS
                STATUS
                SUMMARY ] [TO file-id]
```

where

CHARACTERISTICS	displays parameters currently set for the executor, line, or circuit.
STATUS	displays the availability of the executor, line, or circuit for network activity.
SUMMARY	displays a summary of information for the executor, line, circuit, or links.
TO <i>file-id</i>	specifies the name of an output file to which information is to be directed.

Example

The following command displays the permanent characteristics that you can set for your executor node. A sample system response follows:

```
NCP> LIST EXECUTOR CHARACTERISTICS Return
Executor Characteristics as of 23-Aug-1987 16:17:53

Executor node           = 2.975 (ROCKY)

Executor state          = On
Executor Identification = DECnet-DOS V3.0
Node name               = ROCKY
Node address            = 2.975
Incoming timer          = 45
Outgoing timer          = 60
Confidence timer        = 15
```


LIST EXECUTOR

Incoming proxy	= Disabled
Outgoing proxy	= Enabled
Maximum links	= 8
Delay factor	= 16
Delay weight	= 2
Inactivity timer	= 30
Retransmit factor	= 6
Maximum buffers	= 16
Buffer size	= 1498
Segment buffer size	= 557
Receive password	= (password is not set)
Transmit password	= (password is not set)
Transmit pipe quota	= 8
Receive pipe quota	= 3
Out-of-order nak quota	= 0
Autoboot	= Enabled

LIST LINE

The LIST LINE command displays the permanent parameters that can be changed for the DECnet line ASYNC-1 or ETHER-1.

This command displays information that you can change. To view all the information about the line, use the SHOW LINE command.

Format

```
LIST LINE [line-id] [ CHARACTERISTICS
                        STATUS
                        SUMMARY ] [TO file-id]
```

where

<i>line-id</i>	identifies the line (ASYNC-1 or ETHER-1) for which you want information displayed. Because one line is used for each configuration, the <i>line-id</i> is optional.
CHARACTERISTICS	displays parameters currently set for the executor, line, or circuit.
STATUS	displays the availability of the line for network activity.
SUMMARY	displays a summary of information for the line.
TO <i>file-id</i>	specifies the name of an output file to which information is to be directed.

Example

The following command displays permanent information about your line.

```
NCP> LIST LINE CHARACTERISTICS Return
Line Characteristics as of 23-Mar-1989 16:17:57

Line = ETHER-1

Line state                = On
Receive buffers            = 8
NCP>
```


LIST NODE

LIST NODE

The **LIST NODE** command displays permanent local information about a specific remote node.

If you are checking the status of a particular node, that node's name and address must already be defined in the node database. If there is no node database, use the **NCP DEFINE NODE** command to establish a database.

The **LIST NODE** *area.** command displays all the nodes in the local database in one specified area.

The **LIST KNOWN NODES** command displays all the nodes in the local node database.

Format

LIST NODE { *node-id*
*area.** } [TO *file-id*]

or

LIST KNOWN NODES { MS-NET
LAT } [TO *file-id*]

where

node-id specifies the name or the address of the node for which to display information. A node name is a 1- to 6-character alphanumeric string that identifies a specific node. A node address is a numeric character string consisting of an area number and a specific node number. The node address also serves as a unique identification of the node in the network.

TO *file-id* specifies the name of an output file to which information is to be directed.

MS-NET indicates those nodes that have been defined as MS-NET servers.

LAT indicates those nodes that have been defined as LAT host nodes.

Examples

The following command displays information for the node RM5J.

NCP> LIST NODE RM5J Return

Here is a sample system response:

Known permanent nodes as of 20-Feb-1989 16:36:30

Node Address	Node Name	Active Links	LAT/ MS-NET	Account Information
4.30	RM5J	1	M L	

The following command displays the nodes that have been defined in your network database.

NCP> LIST KNOWN NODES Return

Here is a sample system response:

Known permanent nodes as of 20-Feb-1989 16:36:30

Executor node = 2.975 (ROCKY)

Executor state On

Executor Identification DECnet-DOS V3.0

Node Address	Node Name	Active Links	LAT/ MS-NET	Account Information
2.375	BLD3	0		/SMITH/...
4.30	PRSDNT	1	M L	
4.60	BASBAL	0		
4.62	PONTI	0		
4.64	AGNES	0		
4.215	RM10	0		
4.216	RM1A	0		
4.298	DEMBO	0		
55.86	FLR3	0		/D JONES
55.124	BLDJ	1	M L	
55.170	FLR5	0		
55.242	BLDK	0	L	
55.261	RM110	0		
55.335	RM5J	0		

LIST NODE

The following command displays all the nodes that have been defined for area 4 and have been designated as either MS-NET servers or LAT hosts.

NCP> LIST NODE 4.* MSNET LAT Return

Here is a sample system response:

Known Permanent Nodes as of 25-Feb-1989 14:06:25

Executor node = 2.975 (ROCKY)

Executor state On

Executor Identification DECnet-DOS V3.0

Node Address	Node Name	Active Links	LAT/ MS-NET	Account Information
4.30	PRSDNT	0	M L	
4.60	BLDJ	0	M L	
4.298	BLDK	0	L	

LIST OBJECT

The **LIST OBJECT** command displays information about individual objects contained in the Job Spawner object database (DECOBJ.DAT).

The **LIST KNOWN OBJECTS** command displays all the objects contained in the object database.

The **LIST** command is case sensitive. This means that you must specify the object to display by enclosing the name in quotation marks. If you have objects that have been stored in both all capital letters and in mixed-case letters (such as "FAL" and "tell"), you must specify the object exactly as it has been stored in the database.

Format

LIST OBJECT { *object-name*
 object-number } [TO *file-id*]

or

LIST KNOWN OBJECTS [TO *file-id*]

where

object-name specifies the name of the object for which you want to list information. The name can be from 1 to 15 characters long.

object-number specifies the number of the object for which to list information. The value for this number can be from 0 to 255.

TO *file-id* specifies the name of an output file to which information is to be directed.

LIST OBJECT

Examples

The following command displays information in DECOBJ.DAT about the object FAL.

```
NCP> LIST OBJECT FAL Return
```

```
DECnet Objects
  Taskname  File "Arguments"
17  FAL      FAL
```

The following command displays information about all the objects in DECOBJ.DAT.

```
NCP> LIST KNOWN OBJECTS Return
DECnet Objects
  Taskname  File "Arguments"
0    tell   tells
17   FAL    FAL
63   DTR    DTR
```

The following command displays information about the object in the Spawner database.

```
NCP> LIST OBJECT Return
```

The default object number is 0. Therefore, if no object number is requested, all objects of number 0 are displayed.

```
DECnet Objects
  Taskname  File "Arguments"
0    tell   tells
```

LIST REMOTE-ADAPTER-NAME

The **LIST REMOTE-ADAPTER-NAME** command displays information about a specific remote adapter name in the **DECREM.DAT** database. The remote adapter name identifies a network application that is written for the **NETBIOS** emulator interface. The application is a server on a remote **DECnet** system (for example, **VMS services for MS-DOS®**).

The **LIST KNOWN REMOTE-ADAPTER-NAMES** command displays information about all the remote adapter names in the **DECREM.DAT** database.

The **LIST** command for adapter name information is case sensitive. This means that you must specify the exact name to display by enclosing the name in quotation marks. If you have two remote adapter names that use the same letters but are in different cases, such as **"BEST"** and **"best"**, you must be very specific when using the **LIST REMOTE-ADAPTER-NAME** command. If you do not use quotation marks with the **LIST** command, **NCP** displays the name that is stored in all capital letters.

Format

LIST REMOTE-ADAPTER-NAME *name*

or

LIST KNOWN REMOTE-ADAPTER-NAMES [**TO** *file-id*]

where

name is a 1- to 16-character alphabetic string that specifies the remote application for which to display information.

TO *file-id* specifies the name of an output file to which information is to be directed.

LIST REMOTE-ADAPTER-NAME

Example

The following command displays information about the remote adapter named WINTER.

```
NCP> LIST REMOTE-ADAPTER-NAME Winter Return
```

Permanent Remote-Adapter database as of 9-Mar-1989 9:35:59

Remote-Adapter	Node Name	Node Address	Obj #	Access
WINTER	BLD3	10.497	24	

NCP>

LOOP

The **LOOP** command performs loop tests for the DECnet hardware and software. You can use the loop tests for Ethernet configurations.

LOOP tests can be performed on the following levels:

- **CIRCUIT**—performs loop tests over the DECnet circuit.
- **EXECUTOR**—performs loop tests for your local executor node.
- **LINE**—performs loop tests to a loopback plug on the Ethernet controller.
- **NODE**—performs loop tests for the remote node specified.

LOOP CIRCUIT

LOOP CIRCUIT

The **LOOP CIRCUIT** command performs loop tests for the DECnet circuit. You can use the loop circuit tests for either asynchronous DDCMP configurations or for Ethernet configurations.

If you are using DDCMP, the tests verify that network communication is possible between your local node and the following:

- The loopback plug or controller.
- The loopback plug or cable.
- The local modem.
- The remote modem.
- The adjacent node.

If you are using Ethernet, the tests verify that network communication is possible between your local node and the following:

- A specific remote node.
- Any remote node that responds to the loopback request.

When you issue the **LOOP CIRCUIT** command, NCP determines whether you are using DDCMP or Ethernet on your node. The prompts that NCP returns for this command will vary depending upon your node's configuration.

When you run the **LOOP CIRCUIT** test for DDCMP, NCP prompts you for information needed to run the test. You need to perform one of the following operations:

- Place the loopback plug on the controller.
- Place the loopback plug on the end of the cable that is attached to the controller.
- Place the local modem in loopback mode (by pressing a switch on the modem).
- Place the remote modem in loopback mode (by pressing a switch on the modem).

LOOP CIRCUIT

- If you are performing a loop circuit test to an adjacent node, make sure the adjacent node is set with **CIRCUIT STATE SERVICE**. (Note that this is necessary for DDCMP configurations.)

When you run the **LOOP CIRCUIT** test for Ethernet, you do not need to perform these operations. If you issue the command without specifying a responding node, a loop request is sent out to the multicast loopback assistance address. This means that any node on the Ethernet can reply to the loop request. After a node responds to this request, NCP displays information about the loop message that was sent and the number of the responding node.

When you run any of the **LOOP CIRCUIT** tests, they check the network path from the local node to the hardware level on the remote node, the modem, or the loopback plug.

NCP performs each test n times, as long as the test is successful. The default value for n is 1. You can change this value by using the **COUNT n** option after the **LOOP** command.

When the test is complete, NCP displays a success message. If the test fails, NCP displays an error message.

Format for DDCMP Configurations

LOOP CIRCUIT [COUNT n] $\left[\text{WITH} \left\{ \begin{array}{c} \text{MIXED} \\ \text{ONES} \\ \text{ZEROES} \end{array} \right\} \right]$ [LENGTH n]

where

COUNT n is the number of times to repeat the loop test. The default is 1.

MIXED
ONES
ZEROES indicates that the message type should be mixed ones and zeroes, all ones, or all zeroes. The default is **MIXED**.

LENGTH n is the length of the loop message to be sent. The maximum value you can use is 512. The default is 40.

LOOP CIRCUIT

Format for Ethernet Configurations

LOOP CIRCUIT { **PHYSICAL-ADDRESS** *address* } [**COUNT** *n*] [**WITH** { **MIXED**
ONES
ZEROES }] [**LENGTH** *n*]
NODE *node-id*

where

COUNT <i>n</i>	is the number of times to repeat the loop test. The default is 1.
MIXED ONES ZEROES	indicates that the message type should be mixed ones and zeroes, all ones, or all zeroes. The default is MIXED.
LENGTH <i>n</i>	is the length of the loop message to be sent. The maximum value you can use is 512. The minimum value is 46. This is also the default.
NODE <i>node-id</i>	is the name of the designated node to which to send loop messages. You can use either the node's name or its address. The node must already be defined with the NCP command, SET NODE.
PHYSICAL-ADDRESS <i>address</i>	is a hexadecimal number (hh-hh-hh-hh-hh-hh) representing the actual physical address of the node you are trying to contact on the local Ethernet. The range for this number is 0 - F.

Examples

The following example illustrates a DDCMP loop circuit test.

NCP> LOOP CIRCUIT Return

LOOP CIRCUIT (DDCMP) - Do one of the following, then press any key to begin test. Check the Network Management Guide for details.

1. Place loopback plug on controller.
2. Place loopback plug on end of cable attached to controller.
3. Place local modem in loopback.
4. Place remote modem in loopback.

LOOP CIRCUIT

5. For circuit loopback to the adjacent node, be sure the adjacent node is set with CIRCUIT STATE SERVICE.

The following example illustrates an Ethernet loop circuit test to a multicast address.

```
NCP> LOOP CIRCUIT Return
```

```
LOOP CIRCUIT test started at 21-Aug-1989
Sending loop message 1, 46.
Message echoed by remote circuit loopback 1, 46 bytes.
Multicast Loop Circuit echoed by node 55.62
LOOP CIRCUIT test finished successfully at 21-Aug-1989 16:26:32
NCP>
```

The following example illustrates an Ethernet loop circuit test to the node 55.101 with a count of 3.

```
NCP> LOOP CIRCUIT NODE 55.101 COUNT 3 Return
```

```
LOOP CIRCUIT test started at 21-Aug-1989 16:29:08
Sending loop message 1, 46.
Message echoed by remote circuit loopback 1, 46 bytes.
Sending loop message 2, 46.
Message echoed by remote circuit loopback 2, 46 bytes.
Sending loop message 3, 46.
Message echoed by remote circuit loopback 3, 46 bytes.
LOOP CIRCUIT test finished successfully at 21-Aug-1989 16:29:43
NCP>
```


LOOP EXECUTOR

LOOP EXECUTOR

The LOOP EXECUTOR command invokes the loop test for your local (executor) node. It verifies the operation of your local node by checking the local network software. This test verifies that your local node software has been properly installed.

When you issue this command, NCP performs the test *n* times, as long as the test is successful. The default value for *n* is 1. You can change this value by using the COUNT *n* option after the LOOP command.

Format

LOOP EXECUTOR COUNT *n* WITH { MIXED
ONES
ZERES } LENGTH *n*

where

COUNT <i>n</i>	is the number of times to repeat the loop test. The default is 1.
MIXED ONES ZERES	indicates that the message type should be mixed ones and zeroes, all ones, or all zeroes. The default is MIXED.
LENGTH <i>n</i>	is the length of the loop message to be sent. The maximum value you can use is 512. The default is 46.

Example

The following example requests that the loop test be performed 10 times, with mixed ones and zeroes and a message length of 50.

```
NCP> LOOP EXECUTOR COUNT 10 WITH MIXED LENGTH 50 Return
```

LOOP EXECUTOR

LOOP NODE test started at 9-Mar-1989 9:39:05

Connect complete to node ROCKY

Remote node maximum buffer size for loopback: 512

Successful send and receive, message 1.

Successful send and receive, message 2.

Successful send and receive, message 3.

Successful send and receive, message 4.

Successful send and receive, message 5.

Successful send and receive, message 6.

Successful send and receive, message 7.

Successful send and receive, message 8.

Successful send and receive, message 9.

Successful send and receive, message 10.

LOOP NODE test finished successfully at 9-Mar-1989 9:39:08

NCP>

LOOP LINE CONTROLLER

LOOP LINE CONTROLLER

The LOOP LINE CONTROLLER command performs a loopback test within the local node at the device (or controller) level.

The LOOP LINE CONTROLLER test requires the use of the loopback connector. You must attach this connector to the Ethernet controller in order to perform this test.

When you issue the LOOP LINE CONTROLLER command, NCP prompts you for information needed to run the test. There are no arguments or switches associated with this command.

This command is not supported on the 3Com EtherLink (3C503).

Format

LOOP LINE CONTROLLER

Example

The following example illustrates a sample loop line controller test and possible system response.

```
NCP> LOOP LINE CONTROLLER Return
```

```
LOOP LINE CONTROLLER (Ethernet)
```

1. Unplug network cable from Controller on back of your computer.
2. Place the loopback plug on the Controller. Test will fail if loopback plug is not in place.
3. Press any key to begin test.
4. At the end of the test, reconnect the Controller to the network.

```
LOOP LINE CONTROLLER test started at 19-Aug-1989 11:08:47
```

```
LOOP LINE CONTROLLER test finished successfully at 19-Aug-1989 11:08:49
```

```
Please remove loopback plug and reconnect your node to the network.
```

```
Press any key to continue.
```

```
NCP>
```

LOOP NODE

The LOOP NODE command invokes the loop test to the remote node. It verifies that your node can communicate with a specified remote node. To perform this test, you need to include any network parameters necessary for accessing the loopback mirror on the remote node. These parameters include the node ID, the user ID, the password, and the account.

It is not necessary to include access information for all systems. Most systems do not require this information for performing the remote node test.

When you issue this command, NCP performs the test *n* times, as long as the test is successful. The default value for *n* is 1. You can change this value by using the COUNT *n* option after the LOOP command.

Before running the test, LOOP NODE sets the line and circuit states to ON. When the test is complete, NCP displays a success message. If the test is unsuccessful, NCP displays an error message.

If you do not specify user information for this command, NCP will use the default information that was set with the DEFINE NODE command.

Format

LOOP NODE *node-id* {

 USER *user-id*

 PASSWORD *password*

 ACCOUNT *account*

 COUNT *n*

 WITH { **MIXED**

 ONES

 ZEROES }

 LENGTH *n*
 }

where

node-id specifies the name or the address of a remote node. A node name is a 1- to 6-character alphanumeric string that identifies a specific node. A node address is a numeric character string consisting of an area number and a specific node number. The node address also serves as a unique identification of the node in the network.

LOOP NODE

<i>user-id</i>	is the user information you can use for accessing the loopback mirror on the remote node. A user ID consists of 1 to 39 alphabetic characters.
<i>password</i>	is a unique character string that you can use for accessing the loopback mirror on the remote node. A user's password consists of 1 to 39 alphanumeric characters.
<i>account</i>	is alphanumeric information for accessing the adjacent node's loopback mirror. The information can consist of 1 to 39 alphanumeric characters.
COUNT <i>n</i>	is the number of times to repeat the loop test. The default is 1.
MIXED ONES ZEROS	indicates that the bits of the message should be all ones, all zeroes, or mixed ones and zeroes. The default is mixed.
LENGTH <i>n</i>	is the length of the loop message to be sent. The default is 46.

Example

The following command specifies the node RM5J as the remote node to be used in the remote node test. It specifies that the loop message should be repeated 5 times with a length of 46.

```
NCP> LOOP NODE RM5J COUNT 5 LENGTH 46 Return
```

LOOP NODE

Here is a sample system response:

```
LOOP NODE test started at 4-Aug-1989 14:32:14
Connect complete to node RM5J
Remote node maximum buffer size for loopback: 4096

Sending loop message 1, 46 bytes.
Successful send and receive, message 1.
Sending loop message 2, 46 bytes.
Successful send and receive, message 2.
Sending loop message 3, 46 bytes.
Successful send and receive, message 3.
Sending loop message 4, 46 bytes.
Sending loop message 5, 46 bytes.
Successful send and receive, message 4.
Successful send and receive, message 5.

LOOP NODE test finished successfully at 4-Aug-1989 14:32:15
```

The messages do not necessarily return before the next message is sent (messages 4 and 5 in this example). This is normal output for the LOOP NODE command.

MIRROR

MIRROR

The MIRROR command is used for LOOP NODE tests from a remote node. MIRROR allows the LOOP NODE test to be run from remote systems that support the NCP loop node test to your local node. The maximum length of a message that NCP will loop back is 512 bytes.

Not all loop tests are available on all remote nodes. The tests you can use depend on the tests that are available at the remote node. The information required to run the test also depends on the remote node you are using.

The MIRROR command starts up a loopback mirror (DECnet object ID 25), that echoes test messages that it receives from a remote node back to that same remote node.

When the test run from the remote node is complete, you can press any key to return to the NCP command level.

Format

MIRROR

Example

The following command starts the loopback mirror on your node.

```
NCP> MIRROR Return
```

NCP displays the following information:

```
Loopback Mirror waiting to echo LOOP NODE test from remote node.  
Press any key to stop Loopback Mirror.
```

If a loop node test from a remote node is detected, NCP displays a message like this one:

```
Connect request received from node 4.299 (FREMI), object R SMITH
```

After you press a key to stop the mirror, NCP displays a message and returns you to the NCP prompt.

```
Loopback mirror stopped  
NCP>
```

MONITOR LOGGING

The MONITOR LOGGING command continually displays network events as they occur. The logging of network events pauses when there are no more events to be displayed. You can stop the display of information logging and return to the NCP prompt by pressing any key on the keyboard.

After the events have been displayed, the information contained in the event buffer is discarded. To retain the logging information, you can redirect it to an output file. This is useful if you wish to store the information in order to analyze it later or if you need to show the information to a network support person. If you specify a file for storing log information, you can stop the input of information into the file by pressing any key.

Format

MONITOR LOGGING [*TO file-id*]

where

TO file-id specifies the name of an output file to which information is to be directed.

Example

The following command displays logging information on your screen.

NCP> **MONITOR LOGGING** Return

Here is a sample system response:

Events logged as of 17-Jun-1989 11:17:41

Event type 4.10 Circuit up
Occurred 17-Jun-1989 11:18:00
Adjacent Node: 55.2 (LOWEND)

Event type 4.10 Circuit up
Occurred 17-Jun-1989 11:18:15
Adjacent Node: 55.2 (LOWEND)

.
.
.

PURGE

PURGE

The PURGE command removes information for the designated user from the permanent databases. The types of information you can purge include:

- **ACCESS**—Deletes all access information for a given node or all known nodes.
- **EXECUTOR**—Deletes specific characteristics of the executor node.
- **NODE**—Deletes known nodes from the permanent database.
- **OBJECT**—Deletes objects from the DECOBJ.DAT database.
- **REMOTE-ADAPTER-NAME**—Deletes remote adapter names from the DECREM.DAT database.

The PURGE command deletes information from the permanent databases. This deletion will not take effect until the next network restart. For the information to be deleted immediately, use the DELETE command.

PURGE ACCESS

The **PURGE ACCESS** command removes incoming access information for the designated user. The access information is stored for each user, and includes the password and access type.

NOTE

If access for a particular user is not defined, the default access is used instead.

The **PURGE KNOWN ACCESS** command removes all incoming access information for all users in the incoming access database.

The **PURGE** command is case sensitive. This means that you must specify the user information in quotation marks if it contains any lowercase letters. If you do not specify the proper format when you issue the **PURGE** command, NCP will not purge the information from the database.

Format

PURGE ACCESS USER *user-name*

or

PURGE KNOWN ACCESS

where

user-name

is a 1- to 39-character alphabetic string that defines the user whose access information you want to remove.

PURGE ACCESS

Examples

The following command removes the entire entry for the user BARKER.

```
NCP> PURGE ACCESS USER BARKER Return
```

The following command removes the entire entry for the user paul.

```
NCP> PURGE ACCESS USER "paul"
```

If you typed the user name paul without quotes, NCP would attempt to purge an entry for PAUL. If one existed, it would be removed; if not, you would receive an error message.

PURGE EXECUTOR

The **PURGE EXECUTOR** command removes specified executor node parameters from the permanent database. The parameters you can remove are **RECEIVE PASSWORD** and **TRANSMIT PASSWORD**. (To remove these parameters from the currently running system, use the **CLEAR EXECUTOR** command.)

The **RECEIVE** and **TRANSMIT** passwords are automatically exchanged when the executor or remote node turns on the communication line between itself and the adjacent node. The network coordinator defines these passwords for DDCMP use. The exchange of passwords also occurs when the communication line is restarted from either node because of errors.

Format

PURGE EXECUTOR { **RECEIVE PASSWORD** }
 { **TRANSMIT PASSWORD** }

where

**RECEIVE
PASSWORD**

removes the password the executor node might receive from the adjacent node in order to exchange messages with the adjacent node. The receive password is used if password checking is enabled on both nodes.

**TRANSMIT
PASSWORD**

removes the password the executor node must transmit to the adjacent node in order to exchange messages with the adjacent node.

NOTE

These passwords are used for asynchronous DDCMP connections. They are used if password checking is enabled on both nodes.

PURGE EXECUTOR

Example

The following command removes the password that the executor node must transmit to the adjacent node when the executor node first begins communications. Until the network coordinator defines a new transmit password, the executor node cannot open the communication line between itself and the adjacent node.

```
NCP> PURGE EXECUTOR TRANSMIT PASSWORD Return
```

PURGE NODE

The **PURGE NODE** command removes parameters stored on the local node for either the specified remote node or all remote nodes known to the local node.

The **PURGE KNOWN NODES** command removes information from the local node permanent database (DECNODE.DAT), including any related user, password, and account information.

Format

PURGE NODE *node-id*

USER
PASSWORD
ACCOUNT
LAT-HOST
MS-NET

or

PURGE KNOWN NODES

where

NODE <i>node-id</i>	specifies the node name (or node address) whose parameters are to be removed.
USER	removes the user-name, password, and accounting information for the specified node.
PASSWORD	removes the password associated with the user-name.
ACCOUNT	removes the accounting information for the specified node.
LAT-HOST	removes the indicator that identifies the node as a LAT host.
MS-NET	removes the indicator that identifies the node as an MS-NET server.

PURGE NODE

Example

The following command removes all parameters for node PARIS from the local node's permanent database.

```
NCP> PURGE NODE PARIS Return
```

PURGE OBJECT

The **PURGE OBJECT** command deletes information in the Spawner database (DECOBJ.DAT) for a specific object. You can delete an object by indicating either the object name or the object number.

The **PURGE KNOWN OBJECTS** command deletes all the information about all the objects in DECOBJ.DAT.

The **PURGE** command is case sensitive. This means that you must specify the object to remove by enclosing the name in quotation marks if the name contains any lowercase letters. If you do not specify the proper object form when you issue the **PURGE** command, NCP will not find the matching object in the database.

Format

PURGE OBJECT { *object-name*
 object-number }

or

PURGE KNOWN OBJECTS

where

<i>object-name</i>	specifies the name of the object for which to delete information. The name can be from 1 to 15 characters long.
<i>object-number</i>	specifies the number of the object for which to delete information. The value for this number can be from 0 to 255.

Example

The following command deletes information about the object FAL.

```
NCP> PURGE OBJECT FAL Return
```


PURGE REMOTE-ADAPTER-NAME

PURGE REMOTE-ADAPTER-NAME

The **PURGE REMOTE-ADAPTER-NAME** command deletes an entry from the permanent remote adapter name database (DECREM.DAT). The remote adapter name identifies a network application that is written for the NETBIOS emulator interface. The application resides on a remote personal computer.

The **PURGE KNOWN REMOTE-ADAPTER-NAMES** command deletes all the remote adapter names (and associated information) from DECREM.DAT.

Format

PURGE REMOTE-ADAPTER-NAME *name*

or

PURGE KNOWN REMOTE-ADAPTER-NAMES

where

name

is a 1- to 16-character alphabetic string that specifies the remote application to remove from the permanent database.

Example

The following command deletes information about the remote adapter named WINSTON.

```
NCP> PURGE REMOTE-ADAPTER-NAME WINSTON Return
```

READ LOG

The READ LOG command displays the contents of the event-logging buffer. After the contents of the buffer have been displayed, the contents of the event-logging buffer are discarded. To retain the logging information, you can redirect it to an output file. This is useful if you wish to store the information in order to analyze it later or if you need to show the information to a network support person.

After the information has been read, it is discarded from the event-logging buffer.

Format

READ LOG [*TO file-id*]

where

TO file-id specifies the name of an output file to which information is to be directed.

Examples

The following command redirects the information in the event-logging buffer to the file LOGFILE.TXT. After the information has been redirected to the output file, it is discarded from the event-logging buffer.

```
NCP> READ LOG TO LOGFILE.TXT Return
```

You can now exit from NCP and display the contents of this particular event buffer by viewing the output file you created. This is useful if you wish to store the information in order to analyze it later or if you need to show the information to a network support person.

The following command displays the contents of the event-logging buffer on the screen.

```
NCP> READ LOG Return
```


READ LOG

Here is a sample system response:

```
Events logged as of 17-Jun-1989 11:17:41
Event type 4.10  Circuit up
Occurred 17-Jun-1989 11:18:00
Adjacent Node: 55.2 (LOWEND)

Event type 4.7   Circuit down - circuit fault
Occurred 17-Jun-1989 11:18:09
Adjacent Node: 55.2 (LOWEND)
Reason: Line synchronization lost

Event type 4.10  Circuit up
Occurred 17-Jun-1989 11:18:15
Adjacent Node: 55.2 (LOWEND)

NCP>
```

To view the contents of LOGFILE.TXT, enter the following command:

```
C:\> TYPE LOGFILE.TXT Return
```

The system responds by displaying the contents of your output file :

```
Events Logged as of 17-Aug-1989 16:46:24

Event type 0.0  Event records lost
Occurred 17-Aug-1989 10:22:48
Records lost: 22

Event type 4.18  Adjacency down - circuit fault
Occurred 17-Aug-1989 10:14:19
Adjacent node: 55.261 (DSRVC1)
Reason: Adjacency address change

Event type 4.15  Adjacency up
Occurred 17-Aug-1989 10:14:19
Adjacent node: 55.152

Event type 4.18  Adjacency down - circuit fault
Occurred 17-Aug-1989 10:18:02
Adjacent node: 55.261 (DSRVC1)
Reason: Adjacency address change

Event type 4.15  Adjacency up
Occurred 17-Aug-1989 10:18:02
Adjacent node: 55.152

NCP>
```

SET

The SET command modifies current information in your DECnet volatile database.

Most parameters are set by your network coordinator. Therefore, you must obtain this information from that person. The values you can change with the SET command are:

- **CIRCUIT**—sets the characteristics of a circuit.
- **ECHO**—allows you to display a redirected input file on the screen while the file is being executed.
- **EXECUTOR**—sets the characteristics of the currently running executor node.
- **LINE**—sets line characteristics of the currently running node.
- **LINK**—terminates logical link connections associated with a specific socket.
- **LOCAL-ADAPTER-NAME**—enters a local adapter name in the volatile database. This identifies a network application residing on your personal computer.
- **PAUSE**—allows you to pause the display if output fills more than one screen.
- **REMOTE-ADAPTER-NAME**—enters a remote adapter name into the volatile database. The remote adapter name identifies a network application residing on a remote node.

This command affects the volatile DECnet databases. The characteristics you change with the SET command will not be saved across network restarts. To save characteristics across network restarts use the DEFINE command.

SET CIRCUIT

SET CIRCUIT

The SET CIRCUIT command modifies current circuit information.

Most parameters are set by your network coordinator. Therefore, you must obtain this information from that person.

Format

SET CIRCUIT [*circuit-id*] {
HELLO TIMER *seconds*
MULTICAST-LISTENER {DISABLED
ENABLED}
OWNER {DECNET
MOP}
SERVICE {DISABLED
ENABLED}
STATE {ON
OFF}}

where

CIRCUIT [*circuit-id*] specifies the circuit, either ASYNC-1 or ETHER-1, for which parameters are to be defined or modified. Because one circuit is used for each configuration, the *circuit-id* is optional.

HELLO TIMER
seconds specifies the frequency of routing hello messages sent to the adjacent node over the circuit. The valid range is 1 to 8191 seconds. The default is 30.

MULTICAST-LISTENER sets the circuit's multicast-listener state. The default is enabled.

DISABLED specifies that the circuit will not respond to multicast requests. This applies to Ethernet communications.

ENABLED specifies that the circuit will respond to multicast requests. This applies to Ethernet communications.

OWNER specifies the circuit owner.

SET CIRCUIT

SERVICE	sets the circuit's service state. The default is ENABLED .
DISABLED	specifies that the circuit is unavailable for service operations.
ENABLED	specifies that the circuit is available for special network activity such as loopback testing.
STATE	sets the line's operational state to ON or OFF . The default state is ON .
ON	allows network traffic to flow over the circuit. This is the normal operational state.
OFF	allows no traffic to flow over the circuit. The circuit is unavailable for any network activity.

Example

The following command identifies the owner of circuit **ETHER-1** as **DECnet**.

```
NCP> SET CIRCUIT ETHER-1 OWNER DECNET Return
```


SET ECHO

SET ECHO

The SET ECHO command determines whether a redirected input file is displayed on your screen as the file is executed. The default is OFF (no echo).

You can use the SET ECHO command within a command file. You cannot use SET ECHO at the NCP prompt (using it this way will have no effect).

Format

SET ECHO { **ON** }
 { **OFF** }

where

ON turns the echo on, causing the file's contents to be displayed on the screen as the file is executed.

OFF turns the echo off.

To use SET ECHO, you should set up a command file that contains the SET ECHO command as well as other NCP commands.

Example

This is a sample command file, named SAMPLE.TXT:

```
SET ECHO ON
SHOW EXECUTOR CHARACTERISTICS
EXIT
```

If you run your SAMPLE.TXT file as a command file, you will see the results of the SHOW EXECUTOR CHARACTERISTICS command on the screen. Then, NCP will exit. For example:

```
C:\> NCP < SAMPLE.TXT Return
NCP>SHOW EXECUTOR CHARACTERISTICS Return

Executor Characteristics as of 23-Oct-1989 16:17:39

Executor node                = 2.975 (ROCKY)
Driver version number        = 3.0.0
Executor state                = On
Executor Identification       = DECnet-DOS V3.0
Executor version              = 3.0.1
Node name                     = ROCKY
Node address                  = 2.975
Incoming timer                 = 45
Outgoing timer                = 60
Confidence timer              = 15
Incoming proxy                 = Disabled
Outgoing proxy                 = Enabled
Active links                   = 3
NSP version                   = 4.0.0
Maximum links                  = 8
Delay factor                   = 16
Delay weight                   = 2
Inactivity timer              = 30
Retransmit factor             = 6
Routing version                = 2.0.0
Routing types                  = Nonrouting IV
Maximum buffers                = 24
Buffer size                    = 1498
Segment buffer size           = 557
Receive password               = (password is not set)
Transmit password              = (password is not set)
Database path                  = C:\DECNET\
Transmit pipe quota            = 8
Receive pipe quota             = 3
PC type                        = IBM PCAT
Out-of-order nak quota        = 0
Autoboot                      = Enabled
NCP> EXIT Return

C:\>
```


SET EXECUTOR

SET EXECUTOR

The SET EXECUTOR command sets the executor node's current parameters or characteristics. These settings are not preserved across network restarts. (For the settings to take effect the next time you boot your system, use the DEFINE EXECUTOR command.)

Format

SET EXECUTOR *ADDRESS node-address*
CONFIDENCE TIMER number
DELAY FACTOR number
DELAY WEIGHT number
IDENTIFICATION id-string
INACTIVITY TIMER seconds
INCOMING TIMER seconds
NAME node-name
NAK QUOTA number
OUTGOING TIMER seconds
RECEIVE PASSWORD password
RECEIVE PIPE QUOTA number
RETRANSMIT FACTOR number
STATE { ON }
{ OFF }
TRANSMIT PASSWORD password
TRANSMIT PIPE QUOTA number

where

ADDRESS
node-address

specifies the address assigned to the executor node. The node address is a unique numeric identification for the node. It must include an area number and a node number. You can set ADDRESS when DNP is installed and the executor's state is OFF. The executor's state is OFF if the DECnet network process fails to find the DECnet database file, DECPARM.DAT or it is defined as OFF.

**CONFIDENCE
TIMER** *number*

is an optional extra amount of time to wait before timing a link out. The value range for this timer is 1 to 65535. The default is 15.

DELAY FACTOR
number

determines the amount of time to wait before a message is retransmitted to a node. The range is 16 to 4095. The default number is 32.

DELAY WEIGHT
number

is the weight to apply to a current round trip delay to a remote node when updating the estimated round-trip delay to a node. The range is 1 to 255. The default is 1.

IDENTIFICATION
id-string

specifies a text identification string for the executor node. The string can consist of 1 to 32 alphanumeric characters. You must use a set of double quotation marks (" ") to delimit any string containing blanks or tabs. You cannot use a single set of quotation marks within the identification string.

You can use the IDENTIFICATION parameter to identify the executor node to other nodes in the network (for example, by including your name or department in the string).

INACTIVITY TIMER
seconds

is the maximum duration of inactivity on a logical link before the node checks to see if the logical link still works. The value range is 1 to 65535. The default is 30.

INCOMING TIMER
seconds

is the maximum duration between the time a connection request is received at the executor node and the time that it takes to accept or reject the request. If the request is not accepted or rejected by the user program within the specified number of seconds, the connection request is automatically rejected. If no value is set, there is no timer. The value range is 1 to 65535. The default is 45.

NAK QUOTA *number*

is the number of out-of-order segments to receive before sending a negative acknowledgment (NAK) out on the logical link. This number must be less than or equal to the RECEIVE PIPE QUOTA. The value range for this quota is 0 to 65535. The default is 0 (or 6 for the 3Com driver).

SET EXECUTOR

NAME *node-name*

specifies the node name for the executor node. The node name is an identification string consisting of one to six alphanumeric characters including at least one alphabetic character. You can set NAME when DNP is installed and the executor's state is OFF.

OUTGOING TIMER
seconds

is the duration between the time that the executor node requests a connection and the time that it takes for the request to be acknowledged at the remote node. The value range is 1 to 32K. The default is 60.

**RECEIVE
PASSWORD**
password

is the password the local node must receive from the adjacent node in order to exchange messages with the adjacent node. The password is a character string consisting of eight alphanumeric characters.

**RECEIVE PIPE
QUOTA** *number*

is the number of segments the network will allow to be received for a link by flow control. The valid range is 1 to 4095. The default is 6.

**RETRANSMIT
FACTOR** *number*

is the maximum number of times that the executor node will restart the retransmission timer when it expires. If the number is exceeded, the logical link is disconnected. The number is a decimal value in the range of 1 to 65535. The default is 12.

STATE

ON—allows logical links to be established to the executor node. You must SET EXECUTOR ADDRESS (and NAME, if desired) before you can SET EXECUTOR STATE ON.

OFF—Deinstalls DNP and stops DECnet-DOS. To restart DECnet-DOS you must rerun DNP. If DNP is loaded in EMS, you cannot set the EXECUTOR STATE OFF. You will receive a "Permission Denied" message.

**TRANSMIT
PASSWORD**
password

is the password the local node must transmit to the adjacent node in order to exchange messages with the adjacent node. The password is a character string consisting of eight alphanumeric characters.

SET EXECUTOR

TRANSMIT PIPE QUOTA *number*

the number of segments the driver will buffer into NSP before it blocks the user on a transmission. (NSP is a Network Services Protocol that NCP uses to perform network management and to exchange messages over logical links.) The valid range is 1 to 65535. The default is 6.

NOTE

Information about RECEIVE PASSWORD and TRANSMIT PASSWORD applies to asynchronous DDCMP communications.

Examples

The following command indicates that the executor node will wait 10 seconds between the time that a connection request is sent and the time it takes for an acknowledgment to be sent by the destination node.

```
NCP> SET EXECUTOR OUTGOING TIMER 10 Return
```

The following command sets the executor node's address to 2.975 and its name to ROCKY.

```
NCP> SET EXECUTOR ADDRESS 2.975 NAME ROCKY Return
```

The following command sets the executor node's state to ON. You must also turn the line state to ON with the SET LINE command. The node can then establish logical links with other nodes.

```
NCP> SET EXECUTOR STATE ON Return
```


SET LINE

SET LINE

The SET LINE command sets line information in the volatile database. The values you set are not preserved across network restarts. (For the settings to take effect the next time you boot your system or restart DECnet, use the DEFINE LINE command.)

Most parameters, such as the baud rate, are set by the person who coordinates activity on your network. Therefore, you must obtain this information from that person.

The executor state must be ON in order for you to define or modify any line parameters. In addition, the line state must be OFF if you wish to specify any parameters other than STATE OFF or RETRANSMIT TIMER. You must execute a SET LINE STATE OFF command before you define or modify any other line parameters. When you are finished, execute a SET LINE STATE ON command to turn the line back on.

Setting the state of the line to OFF in an asynchronous configuration will shut down network processing, but will NOT hang up the phone, even if the circuit's modem parameter is set to full. If you must hang up the phone under program control, use the NCP command SET LINE STATE HANGUP.

Format

SET LINE [*line-id*] {
 DEVICE [*device-id*]
 MODEM { **NULL** }
 FULL }
 SPEED *baud-rate*
 STATE { **OFF** }
 ON }
 HANGUP } }

where

LINE [*line-id*] specifies either ASYNC-1 or ETHER-1 as the line for which parameters are to be created or modified. Because one line is used for each configuration, the *line-id* is optional.

DEVICE *device-id* specifies the device for controlling the line (COM-1, COM-2). The line must be in the OFF state.

MODEM	specifies the type of modem control to be used. Modem control is used with asynchronous DDCMP connections.																
NULL	specifies that modem control signals will not be checked.																
FULL	specifies that modem control signals will be checked. The line will be shut down if the modem loses its connection.																
SPEED <i>baud-rate</i>	sets the DDCMP line speed to one of the following: <table><tr><td>50</td><td>150</td><td>1200</td><td>3600</td></tr><tr><td>75</td><td>200</td><td>1800</td><td>4800</td></tr><tr><td>110</td><td>300</td><td>2000</td><td>7200</td></tr><tr><td>134</td><td>600</td><td>2400</td><td>9600</td></tr></table> <p>The default is 9600. For high performance systems, a speed of 19200 is available. Information about line speed applies to asynchronous DDCMP connections.</p>	50	150	1200	3600	75	200	1800	4800	110	300	2000	7200	134	600	2400	9600
50	150	1200	3600														
75	200	1800	4800														
110	300	2000	7200														
134	600	2400	9600														
STATE	sets the line's operational state to ON, OFF, or HANGUP. For Asynchronous configurations, you must set the line state to off to use the communications port for nonDECnet communications.																

NOTE

Information about modem, line speed, and device applies to asynchronous DDCMP communications.

Examples

The following command sets the speed of line ASYNC-1 to 9600. Line speed is set if you are using asynchronous DDCMP connections. You can change the line speed if the line state is OFF.

NCP> SET LINE ASYNC-1 SPEED 9600 Return

The following command specifies the operational state of the line as off.

NCP> SET LINE STATE OFF Return

SET LINK

SET LINK

The SET LINK command terminates logical link connections associated with a specific socket. The socket is also detached.

The SET KNOWN LINKS command breaks all link connections not set with the socket option, SO_KEEPAIVE, and frees the associated sockets.

To determine what links are active, type SHOW KNOWN LINKS.

A socket is an addressable endpoint of communication within a program or a task. A task uses the socket to send and receive data to and from another task. The SO_KEEPAIVE option keeps any links associated with the socket active, even if you try to disconnect the links. As an example, links established by the Network Device Utility (NDU) for the virtual disk facility are set as KEEPAIVE. Thus, using the SET KNOWN LINKS command would not affect the NDU links.

For more information about sockets and socket options, consult the *DECnet-DOS Programmer's Reference Manual*.

Use the SET LINK or SET KNOWN LINKS commands if you have run out of links and you want to recover without performing a network restart. **You should be very familiar with the purpose of the socket (or sockets) whose state you want to change.**

Format

SET LINK *socket* **STATE OFF**

or

SET KNOWN LINKS STATE OFF

where

LINK *socket* specifies the socket to be detached.

STATE OFF turns the logical link off.

**KNOWN LINKS
STATE OFF**

breaks all current links whose sockets are not set with SO_KEEPAIVE. The sockets can be reassigned to other tasks.

Examples

The following command detaches socket 3 and breaks its logical links.

NCP> SET LINK 3 STATE OFF Return

The following command aborts all current links and frees the associated sockets.

NCP> SET KNOWN LINKS STATE OFF Return

SET LOCAL-ADAPTER-NAME

SET LOCAL-ADAPTER-NAME

The SET LOCAL-ADAPTER-NAME command enters a local NETBIOS name into the volatile database. This information is not preserved across network restarts. (For the settings to take effect the next time you boot your system, use the DEFINE LOCAL-ADAPTER-NAME command.) The local adapter name identifies a network application that is written for the NETBIOS emulator interface. The application resides on your local personal computer.

The SET command for adapter name information is case sensitive. This means that you must specify the name to set by enclosing the name in quotation marks if the name contains any lowercase letters. If you have two local adapter names that use the same letters but are in different cases, such as "BEST" and "best", you must be very specific when using the SET LOCAL-ADAPTER-NAME command.

Format

SET LOCAL-ADAPTER-NAME *name* [BYTE16 *nn*]

where

name is a 1- to 16-character ASCII string that specifies the name of the local application to add to the volatile database.

nn specifies a value for byte 16. This value can be any decimal value from 0 to 255.

Example

The following command enters myapp in the volatile database.

```
NCP> SET LOCAL-ADAPTER-NAME "myapp" Return
```

SET PAUSE

The SET PAUSE command enables or disables display characteristics. If you issue an NCP command that displays several screens of data (such as LIST or SHOW), some of the information might scroll by before you can read it. SET PAUSE lets you control this display of information. For example, if you issue a SET PAUSE command and then issue a LIST command, the display stops at the end of the first screen of data. NCP issues a ~~More~~ prompt for you to continue. When you see this prompt, you can press **Return** to view the rest of the display.

Format

SET PAUSE { ON
 OFF }

Example

The following command turns the PAUSE option ON.

NCP> SET PAUSE ON **Return**

Now when you issue a SHOW EXECUTOR CHARACTERISTICS command, the display stops at the end of the first screen and the ~~More~~ prompt appears. You must then press **Return** before you can view the rest of the display.

SET PAUSE

For example:

```
NCP> SHOW EXECUTOR CHARACTERISTICS Return

Executor Characteristics as of 23-Oct-1989 16:17:39

Executor node           = 2.975 (ROCKY)
Driver version number   = 3.0.0
Executor state          = On
Executor Identification = DECnet-DOS V3.0
Executor version        = 3.0.1
Node name               = ROCKY
Node address            = 2.975
Incoming timer          = 45
Outgoing timer          = 60
Confidence timer        = 15
Incoming proxy          = Disabled
Outgoing proxy          = Enabled
Active links            = 3
NSP version             = 4.0.0
Maximum links           = 8
Delay factor            = 16
Delay weight            = 2
Inactivity timer        = 30
Retransmit factor       = 6
Routing version         = 2.0.0
Routing types           = Nonrouting IV

--More-- Return

Maximum buffers         = 24
Buffer size             = 1498
Segment buffer size     = 557
Receive password        = (password is not set)
Transmit password       = (password is not set)
Database path           = C:\DECNET\
Transmit pipe quota     = 8
Receive pipe quota      = 3
PC type                 = IBM/PCAT
Out-of-order nak quota  = 0
Autoboot                = Enabled
NCP>
```

SET REMOTE-ADAPTER-NAME

The SET REMOTE-ADAPTER-NAME command enters a remote adapter name into the volatile database. This information is not preserved across network restarts. (For the settings to take effect the next time you boot your system, use the DEFINE REMOTE-ADAPTER-NAME command.)

The remote adapter name identifies a network application that is written for the NETBIOS emulator interface. The application resides on a remote personal computer.

The SET command for adapter name information is case sensitive. This means that you must specify the name to set by enclosing the name in quotation marks if the name contains any lowercase letters. If you have two remote adapter names that use the same letters but are in different cases, such as "BEST" and "best", you must be very specific when using the SET REMOTE-ADAPTER-NAME command.

Format

SET REMOTE-ADAPTER-NAME *name* NODE *node-id* OBJECT *number*

USER <i>user-id</i> PASSWORD <i>password</i> ACCOUNT <i>account</i>

where

name

is a 1- to 16-character alphabetic string that specifies the name to assign to the remote application. For the characters representing the name to be stored in uppercase letters, type the characters. If you do not want the characters to be stored as uppercase, place the string within quotation marks.

SET REMOTE-ADAPTER-NAME

<i>node-id</i>	identifies the node associated with this remote adapter. The node you use must be one that is already defined in the node database, DECNODE.DAT. You must identify a node for each remote adapter name that you set. The node ID can be either the node's name or its address. A node name is a 1- to 6-character alphanumeric string that identifies a specific node. A node address is a numeric character string consisting of an area number and a specific node number. The node address also serves as a unique identification of the node in the network.
<i>number</i>	is the number of the DECnet object. The range for this number is 0 to 255. The default is 0.
<i>user-id</i>	is the user information you can use for accessing the loopback mirror on the remote node. A user ID consists of 1 to 39 alphabetic characters.
<i>password</i>	is a unique character string that you use for accessing the loopback mirror on the remote node. A user's password consists of 1 to 39 alphanumeric characters (some systems restrict password lengths to 8 characters).
<i>account</i>	is alphanumeric information for accessing the adjacent node's loopback mirror. The information can consist of 1 to 39 alphanumeric characters.

For the USER, PASSWORD, or ACCOUNT information to be stored in uppercase letters, type the characters. If you do not want the characters to be stored as uppercase, place the string within quotation marks. For example:

abc is stored in the database as **ABC**

"Aab" is stored in the database as **Aab**

Example

The following command enters the REMOTE-ADAPTER-NAME NORMAL in the volatile database.

```
NCP> SET REMOTE-ADAPTER-NAME NORMAL Return
```

SHOW ACTIVE

The SHOW ACTIVE command displays summary type information for all active links between the executor node and the adjacent node, as well as information about the executor node, the adjacent node, and the remote nodes that currently have active links to your local node.

Format

SHOW ACTIVE LINKS [TO *file-id*]

or

SHOW ACTIVE NODES [TO *file-id*]

where

TO *file-id* specifies the name of an output file to which information is to be directed.

Examples

The following command displays a list of all the currently active logical links to your node.

NCP>SHOW ACTIVE LINKS Return

Active Links Status as of 13-Aug-1989 9:42:47

State	Socket	Node	Local Addr	Remote Addr	Local #	Name	Remote #	Name
Running	1	55.124	35392	17455	0	LLA35392	64	
Running	2	4.30	35425	52272	0	LLA35425	64	

NCP>

SHOW ACTIVE

The following command displays a list of all the active nodes that currently have logical links to your node.

```
NCP>SHOW ACTIVE NODES Return
```

```
Active Volatile Nodes as of 13-Aug-1989 9:42:58
```

Node Address	Node Name	Active Links	Delay	Next Node Address	Name
55.124	BLDJ	1			
4.30	ORIOLE	1			

```
NCP>
```

SHOW ADJACENT

On asynchronous nodes, the SHOW ADJACENT command displays information for the adjacent node. On Ethernet nodes, this command displays the designated router.

Format

SHOW ADJACENT NODE [*TO file-id*]

where

TO file-id specifies the name of an output file to which information is to be directed.

Example

This command displays the information for the adjacent node.

NCP> **SHOW ADJACENT NODE** Return

Here is a sample system response:

Known Volatile Nodes as of 21-Aug-1989 16:15:47

Node Address	Node Name	State	Active Links	Circuit ID
55.261	DSRVC1	Reachable	0	ETHER-1

NCP>

SHOW CIRCUIT

SHOW CIRCUIT

The **SHOW CIRCUIT** command displays all the values of the currently running system. You can either display the information on your screen or redirect it to an output file at the local node.

Circuit counters monitor traffic and errors on individual circuits. The network coordinator can periodically check each circuit's counters to assess circuit performance and determine potential problems.

Format

```
SHOW CIRCUIT [circuit-id] [ COUNTERS  
CHARACTERISTICS  
STATUS  
SUMMARY ] [TO file-id]
```

where

CIRCUIT [<i>circuit-id</i>]	specifies the circuit, either ASYNC-1 or ETHER-1 , for which information is to be displayed. Because one circuit is used for each configuration, the <i>circuit-id</i> is optional.
CHARACTERISTICS	displays parameters currently set for the circuit.
COUNTERS	displays error and traffic information for the circuit.
STATUS	displays the availability of the circuit for network activity.
SUMMARY	displays a summary of information for the circuit.
TO <i>file-id</i>	specifies the name of an output file to which information is to be directed.

Examples

The following command displays the current characteristics of circuit ASYNC-1.

```
NCP> SHOW CIRCUIT ASYNC-1 CHARACTERISTICS Return
```

The following command displays the current counters for the circuit.

```
NCP> SHOW CIRCUIT COUNTERS Return
```

The following command redirects the current status information about circuit ETHER-1 to an output file named CIR.DAT.

```
NCP>SHOW CIRCUIT ETHER-1 STATUS TO CIR.DAT Return
```

Circuit counters as of 22-Mar-1989 15:20:23

Circuit = ETHER-1

Seconds since last zeroed	= 1290
Terminating packets received	= 4
Originating packets sent	= 0
Circuit down	= 0
Initialization failure	= 0
Bytes received	= 32020
Bytes sent	= 1452
Data blocks received	= 187
Data blocks sent	= 44
User buffers unavailable	= 0

NCP>

SHOW EXECUTOR

SHOW EXECUTOR

The **SHOW EXECUTOR** command displays the current executor node information. You can either display the information on your screen or redirect it to an output file at the local node.

Format

```
SHOW EXECUTOR [ COUNTERS  
                  CHARACTERISTICS  
                  STATUS  
                  SUMMARY ] [TO file-id]
```

where

CHARACTERISTICS	displays parameters currently set for the executor.
COUNTERS	displays error and traffic information for the executor.
STATUS	displays the availability of the executor, for network activity.
SUMMARY	displays a summary of information for the executor.
TO <i>file-id</i>	specifies the name of an output file to which information is to be directed.

Examples

The following command redirects the status information for the executor node to an output file named **STAT.NOD**.

```
NCP> SHOW EXECUTOR STATUS TO STAT.NOD Return
```

SHOW EXECUTOR

To display this file, you must exit NCP by pressing **Ctrl/Z** **Return**, and then enter the TYPE command:

```
C:\> TYPE STAT.NOD Return  
Executor Status as of 9-Mar-1989 9:47:19  
  
Executor node      = 2.975 (ROCKY)  
  
State              = On  
Executor identification = DECnet-DOS V3.0  
  
NCP>
```

The following command displays the characteristics for the executor node.

```
NCP> SHOW EXECUTOR CHARACTERISTICS Return
```


SHOW LAT

SHOW LAT

The SHOW LAT command displays information about the LAT connection you are using. The information can be directed to a file. This command works if you have LAT installed on your node.

Format

SHOW LAT [COUNTERS] [TO *file-id*]

where

COUNTERS displays error and traffic information for the executor, line, or circuit.

TO *file-id* specifies the name of an output file to which information is to be directed.

Example

The following command displays the counter information for the LAT line.

```
NCP> SHOW LAT COUNTERS Return
```

SHOW LINE

The **SHOW LINE** command displays current line information. You can either display the information on your screen or redirect it to an output file at the local node.

Line counters monitor procedures on individual lines. The network coordinator can periodically check each line's counters to assess line performance and determine potential problems.

Format

```
SHOW LINE [line-id] [ CHARACTERISTICS  
                        STATUS  
                        SUMMARY  
                        COUNTERS ] [TO file-id]
```

where

LINE <i>line-id</i>	specifies either ASYNC-1 or ETHER-1 as the line for which information is to be displayed. Because there is one line for each configuration, the <i>line-id</i> is optional.
CHARACTERISTICS	displays parameters currently set for the line.
COUNTERS	displays error and traffic information for the line.
STATUS	displays the availability of the line for network activity.
SUMMARY	displays a summary of information for the executor, line, circuit, or links.
TO <i>file-id</i>	specifies the name of an output file to which information is to be directed.

SHOW LINE

Example

The following command redirects counter information about line ETHER-1 to an output file named STORE.LIN.

```
NCP> SHOW LINE ETHER-1 CHARACTERISTICS TO STORE.LIN Return
```

If you enter the TYPE command to display its contents, you will see the following:

Line Characteristics as of 22-Mar-1989 15:26:41

Line = ETHER-1

Line state	= ON
Line substate	= Running
Device Id	= LPC-1
Receive buffers	= 8
Protocol type	= Ethernet
Hardware address	= 08-00-2B-09-9C-89
Station Address	= AA-00-04-00-E8-13

NCP>

SHOW LOCAL-ADAPTER-NAME

The SHOW LOCAL-ADAPTER-NAME command displays information about the local adapter name in the volatile database.

The local adapter name identifies a network application that is written for the NETBIOS emulator interface. The application resides on your local personal computer.

The SHOW command for adapter name information is case sensitive. This means that you must specify the name to display by enclosing the name in quotation marks if the name contains any lowercase letters. If you have two local adapter names that use the same letters but are in different cases, such as "BEST" and "best", you must be very specific when using the SHOW LOCAL-ADAPTER-NAME command. If you do not use quotation marks with the SHOW command, NCP displays the name that is stored in all capital letters.

Format

SHOW LOCAL-ADAPTER-NAME *name* [BYTE16 *nn*]

or

SHOW KNOWN LOCAL-ADAPTER-NAMES

where

name is a 1- to 16-character ASCII string that specifies the local application for which to display information.

nn specifies a value for byte 16. This value can be any decimal value from 0 to 255.

Example

The following command displays information about the local adapter named REGAL.

```
NCP> SHOW LOCAL-ADAPTER-NAME REGAL Return
```


SHOW NODE

SHOW NODE

The **SHOW NODE** command displays information about a specific node if you have made a DECnet connection since the last network restart.

The **SHOW KNOWN NODES** command displays information about all the nodes to which you have made DECnet connections since the last network restart. This command always displays the executor node and the adjacent node.

Format

SHOW NODE *node-id*

or

SHOW KNOWN NODES

where

node-id specifies the name or the address of the node for which to display information.

Examples

The following command displays information for the node ORIOLE.

```
NCP>SHOW NODE ORIOLE Return
```

```
Known Volatile Nodes as of 17-Oct-1989 15:12:11
```

Node Address	Node Name	Active Links	Delay	Next Node Address	Name
4.30	ORIOLE	0	3		

```
NCP>
```

SHOW NODE

The following command displays information about nodes that have been accessed through DECnet-DOS since network restart.

NCP> SHOW KNOWN NODES Return

Known Volatile Nodes as of 17-Oct-1989 15:20:33

Node Address	Node Name	Active Links	Delay	Next Node Address	Node Name
2.975	ROCKY				
55.161					
4.298	DELON	0	1		

NCP>

SHOW REMOTE-ADAPTER-NAME

SHOW REMOTE-ADAPTER-NAME

The **SHOW REMOTE-ADAPTER-NAME** command displays information about the remote adapter name in the volatile database.

The remote adapter name identifies a network application that is written for the NETBIOS emulator interface. The application resides on a remote personal computer.

The **SHOW** command for adapter name information is case sensitive. This means that you must specify the name to display by enclosing the name in quotation marks if the name contains any lowercase letters. If you have two remote adapter names that use the same letters but are in different cases, such as "BEST" and "best", you must be very specific when using the **SHOW REMOTE-ADAPTER-NAME** command. If you do not use quotation marks with the **SHOW** command, NCP displays the name that is stored in all capital letters.

Format

SHOW REMOTE-ADAPTER-NAME *name*

or

SHOW KNOWN REMOTE-ADAPTER-NAMES

where

name is a 1- to 16-character alphabetic string that specifies the remote application for which to display information.

Example

The following command displays information about the **REMOTE-ADAPTER-NAME PERSIS** in the volatile database.

```
NCP> SHOW REMOTE-ADAPTER-NAME PERSIS Return
```

TELL

The TELL command instructs a remote DECnet node to display information about its executor, lines, nodes, or circuits.

If the remote system requires user and password information, these must be set using the DEFINE NODE command.

Format

```

TELL node-id SHOW EXECUTOR  $\left[ \begin{array}{l} \text{CHARACTER} \\ \text{STATUS} \\ \text{SUMMARY} \end{array} \right]$  [TO file-id]

TELL node-id SHOW CIRCUIT circuit-id  $\left[ \begin{array}{l} \text{CHARACTER} \\ \text{STATUS} \\ \text{SUMMARY} \end{array} \right]$  [TO file-id]

TELL node-id SHOW KNOWN CIRCUIT  $\left[ \begin{array}{l} \text{CHARACTER} \\ \text{STATUS} \\ \text{SUMMARY} \end{array} \right]$  [TO file-id]

TELL node-id SHOW LINE line-id  $\left[ \begin{array}{l} \text{CHARACTER} \\ \text{STATUS} \\ \text{SUMMARY} \end{array} \right]$  [TO file-id]

TELL node-id SHOW KNOWN LINES  $\left[ \begin{array}{l} \text{CHARACTER} \\ \text{STATUS} \\ \text{SUMMARY} \end{array} \right]$  [TO file-id]

TELL node-id SHOW KNOWN NODES [TO file-id]

TELL node-id SHOW AREA number [TO file-id]

TELL node-id SHOW KNOWN AREA [TO file-id]

TELL node-id SHOW NODE  $\left\{ \begin{array}{l} \text{node-id} \\ \text{*node-id} \\ \text{area.*} \end{array} \right\}$  [TO file-id]

```

or

```

TELL node-id ZERO  $\left\{ \begin{array}{l} \text{EXECUTOR} \\ \text{CIRCUIT } \textit{circuit-id} \\ \text{LINE } \textit{line-id} \end{array} \right\}$  [COUNTER] [TO file-id]

```

where

CIRCUIT *circuit-id* specifies the circuit on the remote node for which information is to be displayed.

TELL

LINE <i>line-id</i>	specifies the line on the remote node for which information is to be displayed.
<i>number</i>	specifies a certain area number for which you want to view node information. The value of <i>number</i> can be from 1 to 63. SHOW KNOWN AREA is for use with Level 2 routers.
<i>area.*</i>	specifies the particular node area for which to display information. The asterisk (*) is a wildcard that indicates all the node information for the specified area. (The use of wildcards works if the remote system supports wildcarded node names for network management.)
<i>*.node-id</i>	specifies a particular node id, regardless of area. Indicates that you want information on all nodes of that node-id.
CHARACTERISTICS	displays parameters currently set for the executor, line, or circuit.
COUNTERS	displays error and traffic information for the executor, line, or circuit.
STATUS	displays the availability of the executor, line, or circuit for network activity.
SUMMARY	displays a summary of information for the executor, line, circuit, or links.
TO <i>file-id</i>	specifies the name of an output file to which information is to be directed.

Example

The following command instructs the node RM5J to display information about its executor characteristics.

```
NCP> TELL RM5J SHOW EXECUTOR CHARACTERISTICS Return
```

Here is a sample system response:

```
Executor address      = 4.30
Executor name         = RM5J

Driver version number = 3.0.0
State                 = ON
Executor Identification = DECnet-DOS V3.0
Executor Version      = 4 2 0
Incoming timer        = 45
Outgoing timer        = 60
Confidence timer      = 15
Incoming proxy        = 1
Outgoing proxy        = 0
Active links          = 3
NSP version           = 4 0 0
Maximum links         = 16
Delay factor          = 32
Delay weight          = 3
Inactivity timer      = 60
Retransmit factor     = 12
Routing version       = 2 0 0
Routing types         = Endnode IV
Maximum buffers       = 127
Buffer size           = 576
Segment Buffer Size   = 557
Transmit password     =
Receive password      =
Database Path         = C:\DECnet\
Transmit pipe quota   = 6
Receive pipe quota    = 6
PC type               = 7
Nak quota             = 0
Autoboot              = 0
NCP>
```


ZERO

ZERO

The **ZERO** command zeroes the counters for your personal computer's communications. The following counters can be zeroed:

- **CIRCUIT**—Zeroes counters for the personal computer's circuit.
- **EXECUTOR**—Zeroes counters associated with and maintained on the executor node.
- **LAT**—Zeroes counters for the LAT.
- **LINE**—Zeroes counters for the line.

ZERO CIRCUIT

The ZERO CIRCUIT command zeroes the counters for the circuit.

Format

ZERO CIRCUIT [*circuit-id*] [COUNTERS]

where

CIRCUIT [*circuit-id*] specifies either ASYNC-1 or ETHER-1 as the circuit for which counters are to be zeroed.

Example

The following command zeroes the counters for circuit ETHER-1.

```
NCP> ZERO CIRCUIT ETHER-1 Return
```


ZERO EXECUTOR

ZERO EXECUTOR

The **ZERO EXECUTOR** command zeroes the counters associated with and maintained on the executor node.

Format

ZERO EXECUTOR [COUNTERS]

Example

The following command zeroes the counters associated with the executor node.

```
NCP> ZERO EXECUTOR COUNTERS Return
```

ZERO LAT

The ZERO LAT command zeroes the counters for the LAT line. This command works if you have LAT installed on your node.

Format

ZERO LAT [COUNTERS]

Example

The following command sets the LAT counters to zero.

```
NCP> ZERO LAT COUNTERS Return
```


ZERO LINE

ZERO LINE

The ZERO LINE command zeroes the counters for the line.

Format

ZERO LINE [*line-id*] [COUNTERS]

where

LINE [*line-id*] specifies either ASYNC-1 or ETHER-1 as the line for which counters are to be zeroed. Because there is one line for each configuration, the *line-id* is optional.

Example

The following command zeroes the counters for line ETHER-1.

NCP> ZERO LINE ETHER-1 Return

NCP Error Messages

A.1 Introduction

Some common NCP error messages are listed on the following pages. An explanation and the appropriate user response are also included with each error message.

A.2 Types of NCP Error Messages

NCP error messages fall into three categories:

- **Command line errors.** These messages are displayed when you use the incorrect syntax for a command line.
- **File input/output (I/O) errors.** These messages are displayed when you try to access MS-DOS files.
- **Network errors.** These messages are displayed when you try to run NCP on a system where network support, or some part of it, is missing or not working properly.

A.3 Command Line Errors

The messages listed in this section are all related to errors in command line entry.

Cannot set DDCMP parameter in Ethernet configurations.

You are running DECnet-DOS over an Ethernet connection, and you tried to set a parameter used only by DDCMP asynchronous lines.

Action: Use a different parameter, or reconfigure your connection for Ethernet communication.

Cannot set Ethernet parameter in DDCMP configurations.

You are running DECnet-DOS on an asynchronous line, and you tried to set a parameter used by Ethernet.

Action: Use a different parameter, or reconfigure your connection for DDCMP communications.

Could not find node address with node name = *node-name*.

The node address and node name pair did not match.

Action: Check your list of node names and node addresses and attempt to execute the NCP command again.

Could not find node name with node address = *node-address*.

NCP was unable to find the node name defined by the specific node address.

Action: Check your list of node names and node addresses and try to execute the NCP command again.

Executor is in wrong state.

The executor is not in the proper state for changing the parameter you specified. (For example, you cannot set the EXECUTOR NAME or EXECUTOR ADDRESS when the state is ON.)

Action: Use the DEFINE EXECUTOR command rather than SET EXECUTOR.

Line is in wrong state.

The line is not in the proper state for changing the parameter you specified. (For example, you cannot change the line speed while the line state is ON.)

Action: Change the line state first, then change the parameter.

Node address must be in the range 1 to 1023.

The node address has exceeded the valid range.

Action: Enter a node address that is in the range of 1 to 1023.

Node area must be in the range 1 to 63.

The area number has exceeded the valid range.

Action: Enter a node area that is in the range of 1 to 63.

Node name already defined for another node number.

You tried to define a node name that has already been matched to a different node number.

Action: Select another name to apply to this node number.

Node name must contain an alphabetic character.

The node name you defined did not contain any alphabetic characters. Node names can consist of 1 to 6 alphanumeric characters, but at least one character must be a letter.

Action: Redefine the node name using at least one letter.

Node number already defined for another node name.

You tried to define a node number that has already been matched to a different node name.

Action: Select another number to apply to this node name.

Value out of range; check the documentation or type HELP.

The value you selected for this parameter is not within the valid range.

Action: The values for NCP parameters are defined in Chapter 2. Typing HELP will also provide the valid ranges for these parameters. Check the valid range for the parameter you want to define, then enter a correct value.

String too long. Maximum length is *n*.

The string you entered has too many characters in it.

Action: Enter a string that is in the range specified in the message (this message can appear for several different commands).

^User command error

The command line contains an invalid or missing keyword or parameter. The incorrect command line will be displayed with this error message on the next line. The circumflex (^) will point to the portion of the command line that was incorrect.

Action: Request on-line help for the specific command by using the NCP command HELP.

A.4 File Input/Output (I/O) Errors

The messages listed in this section are all related to accessing MS-DOS files.

Could not open file: {file-name}

The file was not found on the default database device in the \DECNET directory or an MS-DOS error occurred while you attempted to create or open the file.

Action: Refer to the appropriate installation guide for details on the \DECNET directory.

Could not read from file: {file-name}

An error occurred while you attempted to read from the file. The file may not have been properly created/opened, an attempt to read past the end of the file may have occurred, or the file may not have been found in the default database device in the \DECNET directory.

Action: Contact the person responsible for your network for assistance.

Could not seek in file: {file-name}

An error occurred while you attempted to modify the current seek position in the file.

Action: Contact the person responsible for your network for assistance.

Could not write to file: {file-name}

An error occurred while you attempted to write data to the file. The file may not have been properly created/opened, or the disk may be full.

Action: Contact the person responsible for your network for assistance.

A.5 Network Error Messages

The messages listed in this section are displayed when you try to run NCP on a system where network support (or some part of it) is missing or not working properly.

Network Error: Can't assign requested address

A request to set up a node name and node address from a remote node has failed. The node name and node address supplied by the user was not recognized by the local or remote node.

Action: You should specify a different node name and node address. Retry the operation.

Network Error: Connection timed out

A request to set up a node name and node address from a remote node has failed. The time period that is allowed for connecting to the remote network management listener has expired.

Action: Set the executor's timeout parameters to longer periods of time. Try the command again.

Network Error: Executor is in wrong state

An attempt to set a specific network parameter has failed. This occurs when you try to set the executor's node name or node address and the executor's state is already ON.

Action: Delete the DECPARM.DAT file in your \DECNET directory. Run the SETUP program and @NCP.TXT file as detailed in the appropriate installation guide.

Network Error: Host is unreachable

A request to set up a node name and node address from a remote node has failed. The remote node cannot be reached.

Action: Try a different remote node.

Network Error: Line is in wrong state

An attempt has been made to set certain line parameters when the line state is already ON.

Action: You must first set the line state to OFF. Then use the SET LINE command to modify certain line parameters. Then turn the line state to ON again.

Network Error: Network is down

An attempt was made to access the network with the executor state OFF.

Action: Verify that the executor's node address (and name) has been set. Set the executor state to ON with the SET EXECUTOR command, and turn the line state to ON with the SET LINE command.

Network Error: Network is unreachable

The DECnet driver was not properly installed.

Action: Refer to the appropriate installation guide for installation instructions. Repeat the installation procedure.

Network Error: No response from object

A request to set up a node name and node address from a remote node has failed. The logical link connection is broken, or the remote network management listener has failed.

Action: Try to execute your network request again. You can increase the executor's timeout period or try to make a connection to a different remote node.

Network Error: No such process

A request to set up a node name and node address from a remote node has failed. The remote node has no network management listener object.

Action: Try another remote node.

Network Error: Permission denied

An attempt was made to turn the executor state to ON when the executor's address was not set. An attempt was made to set the line state to ON when the executor's state is OFF.

Or, an attempt was made to unload the network when DNP was in EMS, or a TSR was loaded into conventional memory after DNP, and the unloading of DNP would leave a memory hole.

Action: Use the SET EXECUTOR command to set the executor's node address, then turn the executor state ON with the SET EXECUTOR command.

Network Error: Socket operation on non-socket

The specified socket number was invalid.

Action: Specify a valid socket number using the SET LINK command. Refer to the *DECnet-DOS Programmer's Reference Manual* for more details.

A.6 Loop Test Error Messages

When you test the network using loop commands, you may see the following error messages in addition to other network error messages.

Please run NCP to SET NODE *n* NAME.

There is no node name defined for performing the loop tests.

Action: Be sure you have defined the proper node name to use for performing loop tests. You can define the name using the NCP command SET NODE *n* NAME.

Cannot access information about executor node. Please check that the DECnet-DOS network driver is installed and that NCP runs to:

SET EXECUTOR ADDRESS *node-address* **NAME** *node-name*

You have not defined the executor node name and address.

Action: Make sure DECnet-DOS is installed and that you can run NCP. Then use the NCP commands to define the executor node name and address and to turn the executor state ON.

1944-1945

1946-1947

1948-1949

Glossary

access control information

Optional security information that you might need to access a remote node.

account

A character string consisting of 1 to 39 alphanumeric characters.

adjacent node

A node to which your node is physically connected by a single line.

alphanumeric

A string of characters that contains letters (alphabetic characters), numbers (numeric characters), or a combination of both. The term *alphanumeric* is a contraction of the words *alphabetic* and *numeric*.

application

A program (other than the operating system) that performs specific functions in order to meet user requirements. Applications available with DECnet-DOS include the Network File Transfer (NFT) utility, the Network Device Utility (NDU), the Network Control Program (NCP), and the SETHOST utility with network virtual terminal services.

buffer

A temporary storage area in a node's memory. Buffers temporarily hold data being transferred to and from the node. The size and the number of buffers determine the amount of data that can be stored.

circuit

The communications path that operates over a physical line connecting two nodes.

communication path

The route through hardware components and a cable that a message takes when sent from one node to another.

counter

A counter displays statistics about the flow of network messages for your node. The counters also record error conditions and accumulate their totals for you.

DECnet

A family of Digital products that extend the capabilities of Digital computers to perform various operations over a network. DECnet provides many standard applications for accessing files and databases that exist on other nodes.

DECnet-DOS

A set of software and hardware components designed for use with the IBM PC/XT and the IBM Personal Computer AT (including a limited set of compatibles). DECnet-DOS allows you to combine the capabilities of the IBM personal computers with DECnet applications to access the network as an end node.

device

A specific name for a disk or diskette that is currently storing data files.

directory

A specific name assigned to a collection of files stored on a disk or a diskette. A directory can also provide a list of the file names that it contains.

end node

A node that can only send and receive information for its own use. An end node cannot receive and then forward information intended for another node. Your personal computer is an end node.

executor node

A node that runs NCP commands. For a personal computer, this is the local node.

file specification

A complete file identification including an optional drive name and path name. The file specification must include at least a file name followed by an optional file type and version number.

host node

A node on the network that your node can access for the purpose of sharing resources and information.

line

The physical line connecting the local node with the adjacent node.

local node

The node that you are currently using to communicate with other nodes.

logging

The process of recording information from an occurrence in the network. Logging is the process that generates a record of the event.

loopback connector

A hardware device that temporarily connects the output portion of a circuit back to the input portion of a circuit for the purpose of testing communication lines. The loopback connector verifies the operation of devices on the same communications link.

loopback test

A test that sends an electronic signal over a hardware or software connection to verify the operation of devices along the communication link. Certain loopback tests require a loopback connector (see **loopback connector**).

network

A group of computers or systems that are connected together and can communicate with each other to share resources and information.

network manager

A person who assigns and updates node names and node addresses. The network manager also provides administrative assistance to persons using the network.

network link

A temporary connection that establishes communication between programs running on different nodes. The SETHOST utility with network virtual terminal services is a program that establishes a network link with another node, enabling you to log in to a remote host system.

node

A computer with the necessary hardware and software to participate in a network with other computers.

node address

A unique numeric character string that identifies a node to other systems in the network.

node name

A 1- to 6-character alphanumeric string (containing at least one letter) that identifies a node to other users in the network.

password

A character string that uniquely confirms your identity to a system.

process name

A character string that identifies one particular program in a group of programs sharing the same system memory.

protocols

Rules or formats that operating systems must follow to conduct effective communications with other computers in a network.

reachable node

A node to which the local node has a usable communications path.

remote node

Any node on the network other than the node you are currently using.

remote node list

A list containing node definitions of other nodes that are available for communication on the same network.

router

A DECnet node that can receive messages and information from one node and forward them to another node.

routing node

A DECnet node that can receive and forward information from one node to another. A routing node can perform other functions that are not limited strictly to routing information.

topology

The physical arrangement of nodes and connecting hardware that makes up the network.

user name

A character string consisting of 1 to 39 alphabetic characters identifying a user at a remote node.

Index

A

Access control information, 2-3 to
2-5, 3-3, 3-7 to 3-8
 defining, 6-15
 displaying, 2-5, 3-2, 4-6, 6-35
 purging, 6-63, 6-67
 specifying, 2-3, 3-8
Account information, 2-4, 3-8
Active nodes, 4-8

C

Circuit
 defining parameters, 6-17
 displaying information, 4-10, 6-37,
 6-94
 loop testing, 6-50
 setting parameters, 6-74
 showing information, 6-94
 zeroing counters, 6-109
Circuit state, 4-10
Circuit substates, 4-10
Command files, 1-5, 6-76
Counters
 defined, 4-2
 displaying, 4-4
 loopback, 5-6

Counters (Cont.)

 zeroing, 5-6, 6-109, 6-110, 6-112

D

DECnet-DOS
 definitions
 see Glossary

E

Event log
 continuous, 6-61
 reading contents of, 6-71
 sample, 6-61, 6-71
Events
 logging of, 4-11
Executor node
 clearing parameters, 6-7
 defined, 1-4
 defining parameters, 6-19
 displaying characteristics, 6-39
 displaying information, 3-3, 4-7,
 6-39
 loop testing, 6-54
 purging parameters, 6-65
 setting information, 6-78

Executor node (Cont.)

- setting parameters, 6-78
- showing characteristics, 6-96
- zeroing counters, 6-110

G

Getting help, 1-8, 6-33

J

Job Spawner

- creating object database, 6-27
- displaying objects, 6-45
- purging objects, 6-69

L

LAT

- showing information, 6-98
- zeroing counters, 6-111

Line

- defining parameters, 6-23
- displaying information, 4-9, 6-41
- looping, 6-56
- setting line speed, 6-23, 6-83
- setting parameters, 6-82
- showing information, 6-99
- zeroing counters, 6-112

Line states, 4-9

Line substates, 4-9

Links

- showing active, 6-91

Logging

- of events, 4-11

Logical links

- and sockets, 6-84
- displaying information, 3-3, 4-7
- setting maximum number, 6-21

N

NCP

- command format, 1-6
- exiting from, 1-9, 6-32
- getting help for, 1-8, 6-33
- running the utility, 1-5
- starting, 1-5 to 1-6

NCP commands, 6-1

- abbreviating, 1-6
- CLEAR, 6-6
- CLEAR EXECUTOR, 6-7
- CLEAR LOCAL-ADAPTER-NAME, 6-9
- CLEAR REMOTE-ADAPTER-NAME, 6-11
- COPY NODE, 6-12
- DEFINE, 6-14
- DEFINE ACCESS, 6-15
- DEFINE CIRCUIT, 6-17
- DEFINE EXECUTOR, 2-2, 3-7, 6-19
- DEFINE LINE, 6-23
- DEFINE NODE, 2-4, 6-25
- DEFINE OBJECT, 6-27
- DEFINE REMOTE-ADAPTER-NAME, 6-30
- EXIT, 6-32
- HELP, 1-8, 6-33
- LIST, 6-34
- LIST ACCESS, 6-35
- LIST CIRCUIT, 6-37
- LIST EXECUTOR, 6-39
- LIST KNOWN ACCESS, 6-35
- LIST KNOWN NODES, 2-5, 3-2, 3-3, 6-42
- LIST KNOWN OBJECTS, 6-45
- LIST KNOWN REMOTE-ADAPTER-NAMES, 6-47
- LIST LINE, 6-41
- LIST NODE, 6-42
- LIST OBJECT, 6-45

NCP commands (Cont.)

LIST REMOTE-ADAPTER-NAME,
6-47
LOOP, 5-1, 6-49
LOOP CIRCUIT, 6-50
LOOP EXECUTOR, 6-54
LOOP LINE CONTROLLER, 6-56
LOOP NODE, 6-57
MIRROR, 6-60
MONITOR LOGGING, 6-61
PURGE, 6-62
PURGE ACCESS, 6-63
PURGE EXECUTOR, 6-65
PURGE KNOWN ACCESS, 6-63
PURGE KNOWN NODES, 6-67
PURGE KNOWN OBJECTS, 6-69
PURGE KNOWN REMOTE-
ADAPTER-NAMES, 6-70
PURGE NODE, 6-67
PURGE OBJECT, 6-69
PURGE REMOTE-ADAPTER-
NAME, 6-70
READ LOG, 6-71
SET, 6-73
SET CIRCUIT, 6-74
SET ECHO, 1-5, 6-76
SET EXECUTOR, 3-7, 6-78
SET KNOWN LINKS, 6-84
SET LINE, 6-82
SET LINK, 6-84
SET LOCAL-ADAPTER-NAME,
6-86
SET PAUSE, 6-87
SET REMOTE-ADAPTER-NAME,
6-89
SHOW ACTIVE, 6-91
SHOW ACTIVE LINKS, 6-91
SHOW ACTIVE NODES, 4-8, 6-91
SHOW ADJACENT, 6-93
SHOW CIRCUIT, 4-10, 6-94
SHOW CIRCUIT COUNTERS, 5-6
SHOW EXECUTOR, 3-3, 4-7, 6-96

NCP commands (Cont.)

SHOW EXECUTOR COUNTERS,
5-6
SHOW KNOWN NODES, 3-3, 4-6,
4-7, 6-102
SHOW LAT, 6-98
SHOW LINE, 4-9, 6-99
SHOW LINE COUNTERS, 5-6
SHOW LOCAL-ADAPTER-NAME,
6-101
SHOW NODE, 4-8, 6-102
SHOW REMOTE-ADAPTER-
NAME, 6-104
TELL, 6-105
ZERO CIRCUIT, 5-6, 6-109
ZERO EXECUTOR, 5-6, 6-110
ZERO LAT, 6-111
ZERO LINE, 5-6, 6-112

NCP databases, 2-6

commands for permanent, 2-6
commands for volatile, 2-7
contents, 2-6
DECACC.DAT, 2-7, 6-15, 6-35
DECALIAS.DAT, 2-3, 2-4, 2-7, 3-7
DECNODE.DAT, 2-7, 6-12, 6-30,
6-67, 6-90
DECOBJ.DAT, 2-7, 6-27, 6-45,
6-46, 6-69
DECPARM.DAT, 2-7, 6-22, 6-24,
6-78, 6-80
DECREM.DAT, 2-7, 6-30, 6-47,
6-70
permanent, 2-6
volatile, 2-6

NCP error messages

see Appendix A

Network Control Program

see NCP

Network information

changing, 3-2, 3-5
displaying, 3-1
updating, 3-2

Network Management Listener, 1-9, 4-11

Network parameters

changing, 3-5

defined, 3-5

NML, 1-9, 4-11

Node address, 2-2, 2-3, 3-7

Node information

defining address, 2-2, 3-7

defining name, 2-2, 3-7

maintaining, 3-1

managing, 2-1

monitoring, 4-11

purging access control information, 6-67

purging node name, 6-67

purging parameters, 6-67

setting, 2-1

address, 2-1

name, 2-1

updating, 3-1

Node name, 2-2, 2-3, 3-7

Nodes

adjacent, 1-2, 6-93

copying access information, 6-12

defined, 1-2

defining, 6-25

displaying information, 6-42

end, 1-2

executor, 1-4

local, 1-4

looping, 6-57

managing, 1-4

remote, 1-4, 2-3, 3-7

routing, 1-2

showing active, 6-91

showing information, 6-102

P

Password, 2-3, 3-8

R

Reading input from a file, 6-76

Redirected input files, 1-5

used with NCP, 1-5, 6-76

Remote node, 2-3, 3-7

displaying information, 4-8

displaying names, 2-5, 3-3, 4-7

naming, 2-3, 3-7

T

Testing the network, 5-1, 5-2, 6-60

U

User name, 2-3, 3-8

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
call 800-DIGITAL

In Canada
call 800-267-6215

In New Hampshire
Alaska or Hawaii
call 603-884-6660

In Puerto Rico
call 809-754-7575

ELECTRONIC ORDERS (U.S. ONLY)

Dial 800-DEC-DEMO with any VT100 or VT200
compatible terminal and a 1200 baud modem.
If you need assistance, call 1-800-DIGITAL.

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: A&SG Business Manager

INTERNATIONAL

DIGITAL
EQUIPMENT CORPORATION
A&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC),
Digital Equipment Corporation, Westminister, Massachusetts 01473

* Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809-754-7575 x2012

HOW TO ORDER A COPY OF THE REPORT

REPORT NO. 100-100000

DATE: 10/1/70
BY: [illegible]
FOR: [illegible]

REPORT NO. 100-100000

DATE: 10/1/70
BY: [illegible]
FOR: [illegible]

REPORT NO. 100-100000

DATE: 10/1/70
BY: [illegible]
FOR: [illegible]

REPORT NO. 100-100000

DATE: 10/1/70
BY: [illegible]
FOR: [illegible]

REPORT NO. 100-100000

DATE: 10/1/70
BY: [illegible]
FOR: [illegible]

REPORT NO. 100-100000

DATE: 10/1/70
BY: [illegible]
FOR: [illegible]

READER'S COMMENTS

What do you think of this manual? Your comments and suggestions will help us to improve the quality and usefulness of our publications.

Please rate this manual:

	Poor				Excellent
Accuracy	1	2	3	4	5
Readability	1	2	3	4	5
Examples	1	2	3	4	5
Organization	1	2	3	4	5
Completeness	1	2	3	4	5

Did you find errors in this manual? If so, please specify the error(s) and page number(s).

General comments:

Suggestions for improvement:

Name _____ Date _____

Title _____ Department _____

Company _____ Street _____

City _____ State/Country _____ Zip Code _____

DO NOT CUT - FOLD HERE AND TAPE



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY LABEL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

digitalTM

**Networks and
Communications Publications**

550 King Street

Littleton, MA 01460-1289



DO NOT CUT - FOLD HERE

digital